ENGENHARIA DE REQUISITOS

Mestrado Profissionalizante BR Telecom/UNB

Contexto

- A crise do software continua
  - aeroporto de Denver
  - OS360
  - serviço de Ambulância de Londres

© KKB/PUC-Rio

A Boa Notícia…

3

"26% dos projetos de software foram bem-sucedidos."

Standish Group, CHAOS Report, 2000

A Má Notícia…

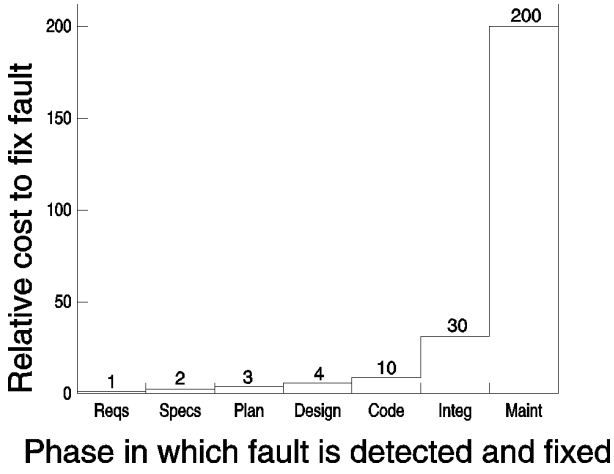
4

Isso significa que 74% falharam!

Standish Group, CHAOS Report, 2000

Schach's Summary I

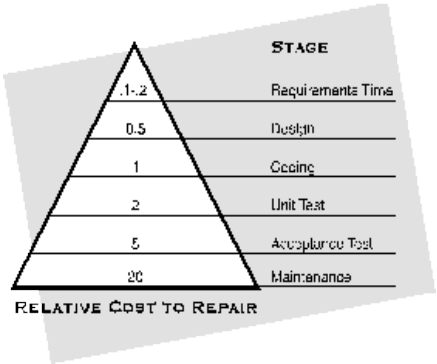
5



© KKB/PUC-Rio

Custo de correção II

6



- **Custo aumenta com o tempo de descoberta do erro**
  - Custo de reparo
  - Custo de perda de oportunidades
  - Custo de perda de clientes
- O custo de **1 problema é 200 vezes** maior se reparado após a implantação.
- Os **erros mais caros** são aqueles cometidos na Análise de requisitos e descobertos pelo usuário!

© KKB/PUC-Rio

Custo de correção III

7

Tom De Marco

8

"*56% dos erros de um software podem ser rastreados para a fase de requisitos*"

© KKB/PUC-Rio

Custo de Erros

- Quanto mais tarde um erro é detectado, maior o custo de corrigí-lo.
- Muitos erros são realizados durante a elicitação e definição de requisitos

© KKB/PUC-Rio

Erros nos requisitos

- Muitos erros nos requisitos podem ser detectados cedo no ciclo de desenvolvimento

Erros no requisitos

11

- Erros típicos incluem fatos incorretos, omissões, inconsistências e ambiguidade.
- Erros nos requisitos constituem uma das maiores preocupações da indústria de software.

© KKB/PUC-Rio

Por que Engenharia de Requisitos?

12

Porquê é difícil?

13

Fonte – Steve Easterbrook

- Problemas tem fronteiras mal definidas (abertos)
- Requisitos estão no contexto organizacional (inclinados a conflitos)
- Soluções para os problemas da análise são artificiais

© KKB/PUC-Rio

Porquê é difícil?

14

Fonte – Steve Easterbrook

- Problemas são dinâmicos
- Requer conhecimento interdisciplinar e habilidades específicas

© KKB/PUC-Rio

Definição

15

"A Engenharia de Requisitos, uma sub-área da Engenharia de Software, estuda o processo de definição dos requisitos que o software deverá atender. A área surgiu em 1993 com a realização do I International Symposium on Requirements Engineering. O processo de definição de requisitos é uma interface entre os desejos e necessidades dos clientes e a posterior implementação desses requisitos em forma de software."

© KKB/PUC-Rio

Objetivos

16

- Entender as necessidades e atender os desejos dos clientes
- Prover ao Engenheiro de Software, métodos, técnicas e ferramentas que auxiliem o processo de compreensão e registro dos requisitos que o software deve atender.

© KKB/PUC-Rio

Outra definição

17

- Engenharia de requisitos é:

"o desenvolvimento e utilização de tecnologia eficaz para a elicitação, especificação e análise dos requisitos dos interessados (clientes/usuários) que devem ser cumpridos por sistemas de software."

© KKB/PUC-Rio

Fred Brook's

18

- Brook's adiciona:

"a parte mais difícil da construção de um sistema de software é decidir, precisamente, o que deve ser construído... Nenhuma outra parte do trabalho aleja mais o sistema resultante se feita errada. Nenhuma outra parte é mais difícil de corrigir depois."

© KKB/PUC-Rio

Processo “essencial”

19

- Entender o problema
  - Utilizar técnicas para elicitar os requisitos: questionários, entrevistas, documentos...
- Modelar o problema
  - Representar nosso entendimento do problemas utilizando técnicas para modelagem: DFD, MER, casos de uso...
- Analizar o problema
  - Verificar e Validar a informação capturada

© KKB/PUC-Rio

Definições

20

- Especificação
  - Descrição minuciosa das caracterítiscas que um material, uma obra, ou um serviço deverão apresentar.

© KKB/PUC-Rio

Definições

21

- Requisito
  - Condição necessária para obtenção de certo objetivo, ou preenchimento de certo objetivo.

© KKB/PUC-Rio

Definições

22

- Requisitos de Software
  - Sentenças que expressam as necessidades dos clientes e que condicionam a qualidade do software.

© KKB/PUC-Rio

Tipos de Requisito

23

- Requisitos Funcionais
  - RF são requisitos diretamente ligados a funcionalidade do software.

© KKB/PUC-Rio

Tipos de Requisito

24

- Requisitos Não Funcionais
  - RNF são requisitos que expressam restrições que o software deve atender ou qualidades específicas que o software deve ter.

© KKB/PUC-Rio

Tipos de Requisito

25

- Requisitos^{-1} (Requisitos Inversos)
  - RIN estabelecem condições que nunca podem ocorrer.

© KKB/PUC-Rio

Exemplos

26

- O sistema deve prover um formulário de entrada para a entrada dos resultados dos testes clínicos de um paciente. (RF)
- Dependendo do resultado do teste, somente o Supervisor pode efetuar a entrada do resultado do teste de um paciente. (RNF de confidencialidade).

© KKB/PUC-Rio

Exemplos

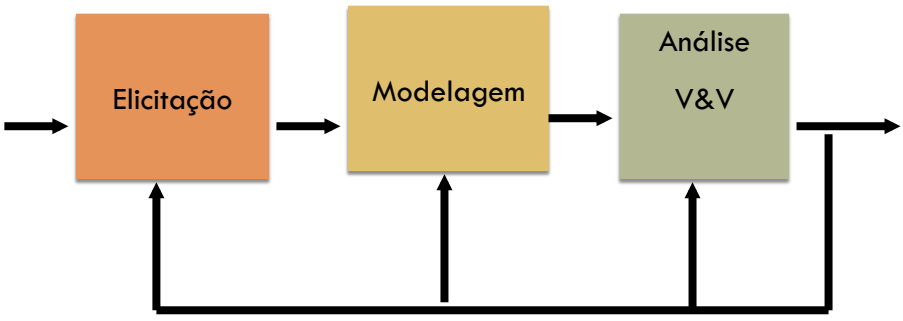
27

- O sistema deve emitir um recibo para o cliente, com o tempo máximo de 8 segundos após a transação. (RF “,” RNF de performace).
- O sistema não pode apagar informação de um cliente (RIN).

© KKB/PUC-Rio

Processo

28



© KKB/PUC-Rio

29

Aspectos ecológicos

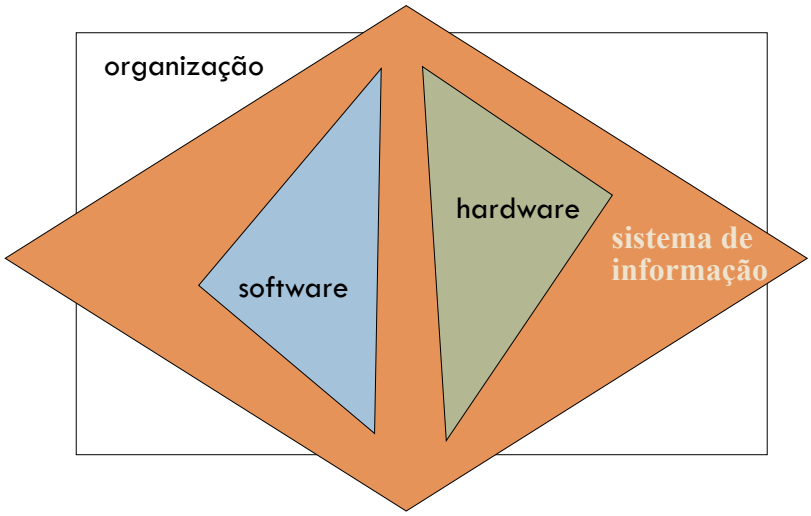
© KKB/PUC-Rio

Definições

30

- **Universo de Informações**
  - É o contexto no qual o software deverá ser desenvolvido e operado. O Udl inclui todas as fontes de informação e todas as pessoas relacionadas ao software. Essas pessoas são também conhecidas como os atores desse universo. O Udl é a realidade circunstanciada pelo conjunto de objetivos definidos pelos que demandam o software.

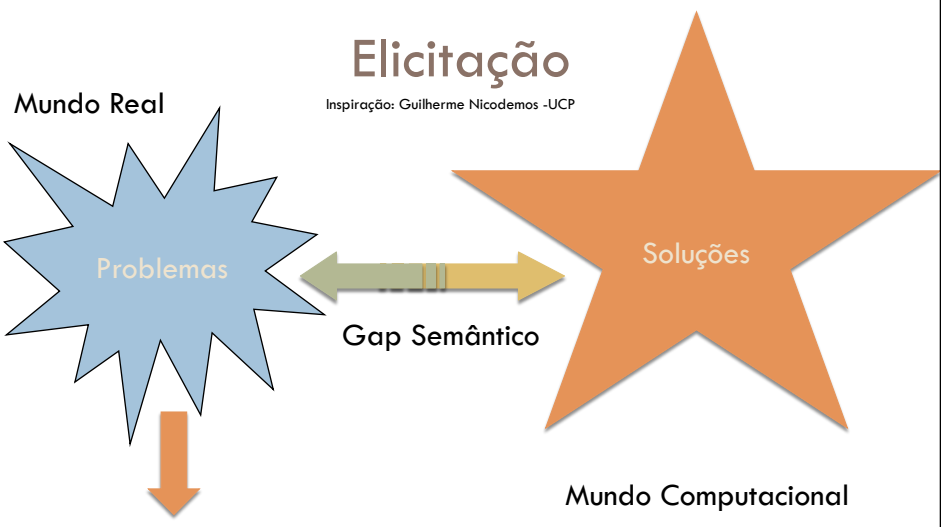
© KKB/PUC-Rio



© KKB/PUC-Rio

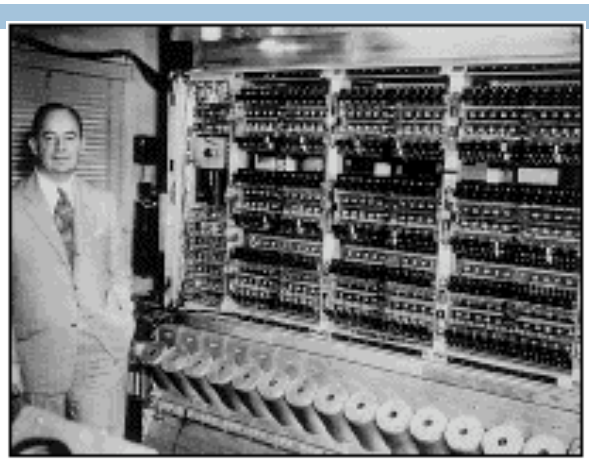
Elicitação de requisitos

© KKB/PUC-Rio



Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio

von Neumann



If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is.

Necessidades da elicitação

35

(perguntas formuladas por Polya)

- O que é desconhecido?
- Você conhece um problema relacionado?
- Você é capaz de reformular o problema?

© KKB/PUC-Rio

Elicitação

36

Elicitar [Var. eliciar + clarear + extrair]
1. descobrir, tornar explícito, obter o máximo de informações para o conhecimento do objeto em questão.

© KKB/PUC-Rio

Elicitação

37

- Identificar as fontes de informação
- Coleta de fatos
- Comunicação

© KKB/PUC-Rio

Identificação das Fontes de Informação

38

- Atores do Universo de Informações
  - Clientes
  - Usuários
  - Desenvolvedores

© KKB/PUC-Rio

Identificação das Fontes de Informação

39

- Documentos
- Livros
- Sistemas de Software
- COTS

© KKB/PUC-Rio

Critérios

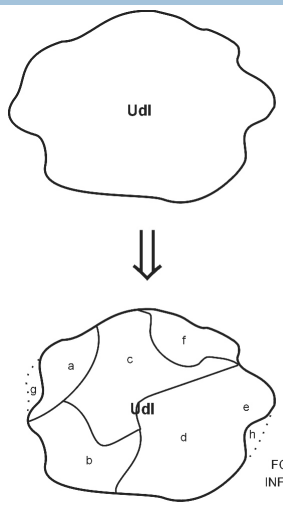
40

- Experiência
- Conhecimento do domínio
- Volume de investimento
- Representatividade
- Função

© KKB/PUC-Rio

Fontes de Informação

41



FONTES DE INFORMAÇÃO
=
{ a, b, c, d, e, f}
U
{ g, h }

© KKB/PUC-Rio

Heurísticas para identificação

42

- Quem é o cliente?
- Quem é o dono do sistema?
- Existe alguma solução (pacote) disponível?

© KKB/PUC-Rio

Heurísticas para identificação de

43

- Quais são os livros relacionados a aplicação em discussão?
- Existe a possibilidade de reutilizar os artefatos (software)?
- Quais são os documentos mais referenciados pelos atores do Udl?

© KKB/PUC-Rio

Coleta de Fatos

44

- Leitura de documentos
- Observação
- Entrevistas
- Reuniões
- Questionários
- Antropologia
- Participação ativa dos atores
- Análise de Protocolos
- Engenharia Reversa
- Reutilização

© KKB/PUC-Rio

Entrevistas

45

- Estruturadas
  - normalmente 1 para 1
  - pode ser do tipo 1 para n ou do tipo n para 1
  - requer algum conhecimento sobre o problema para o preparo das perguntas

© KKB/PUC-Rio

Técnica dos "5" W's

46

- o quê (What?)
- quando (When?)
- onde (Where?)
- por quê (Why?)
- quem (Who?)
- como (How?)

© KKB/PUC-Rio

Entrevista estruturada

47

- O que perguntar
- Como perguntar
- A quem perguntar

© KKB/PUC-Rio

Entrevista não estruturada

48

- Oferece mais flexibilidade
- Informal
- Utilizada durante a fase exploratória

© KKB/PUC-Rio

Conhecimento Tácito

49

- É aquele conhecimento que é trivial para o entrevistado e não o é para o entrevistador

© KKB/PUC-Rio

Conhecimento Tácito

50

Por ser trivial nunca é lembrado como importante e, portanto, não é transmitido ao entrevistador, que, não sabendo, não pode perguntar.

© KKB/PUC-Rio

Entrevistas

51

- \+
  - contato direto com atores
  - possibilidade de validação imediata

© KKB/PUC-Rio

Entrevistas

52

- \-
  - conhecimento tácito
  - diferenças culturais

© KKB/PUC-Rio

Técnicas de leitura

53

- Livros
  - leitura
  - resumos por capítulo
  - evidenciar partes mais importantes
  - utilizar índice por palavra chave
  - vocabulário

© KKB/PUC-Rio

Técnicas de leitura

54

- Documentos do macrosistema
  (leitura mais cuidadosa)
  - sublinhar palavras repetidas
  - sinônimos
  - anotar termos desconhecidos
  - buscar relacionamentos
  - vocabulário

© KKB/PUC-Rio

Técnicas de leitura

55

- Documentos do macrosistema

  procurar entender a ESTRUTURA dos documentos lidos !!!

© KKB/PUC-Rio

Técnicas de leitura

56

- Leitura de documentos similares:
  - identificar estruturas
  - relacionar estruturas
  - vocabulário

© KKB/PUC-Rio

Leitura de Documentos

57

- +
  - facilidade de acesso às fontes de informação
  - volume de informação

© KKB/PUC-Rio

Leitura de Documentos

58

- -
  - dispersão das informações
  - volume de trabalho requerido para identificação dos fatos

© KKB/PUC-Rio

Questionários

59

- Qualitativo
- Quantitativo

© KKB/PUC-Rio

Questionários

60

- O que perguntar
  - exige conhecimento mínimo
  - similar a entrevista estruturada

© KKB/PUC-Rio

Questionários

61

Tipos

- qualitativo
  - possibilita o respondente a abrir o escopo da resposta
  - dificulta a análise posterior
  - perguntas de controle - podemos levar a conflito nas respostas de modo a verificar a consistência do respondente

© KKB/PUC-Rio

Questionários

62

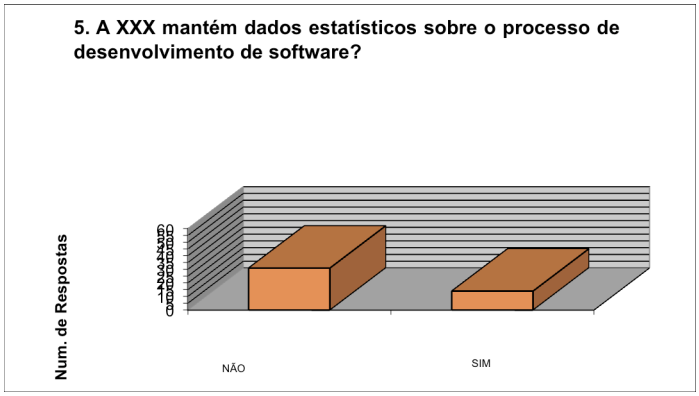
- quantitativo
  - gradação (sim, não / bom, médio,ruim / 0,1,2,3,4)
  - pergunta tem de ser bem elaborada para permitir a distribuição das respostas

© KKB/PUC-Rio

Exemplos

63

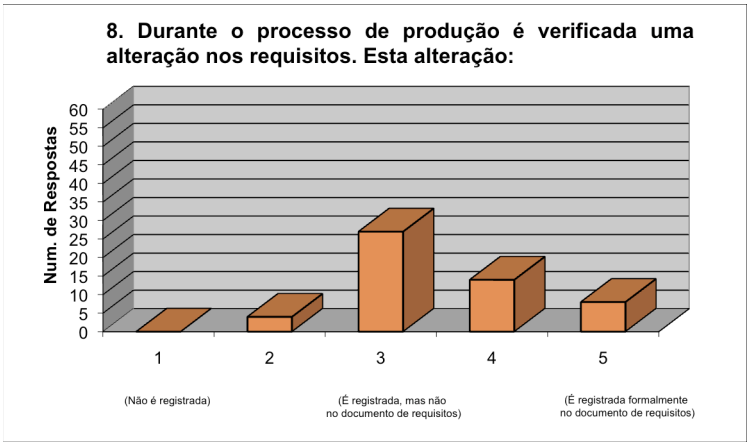
- Quantitativo



© KKB/PUC-Rio

Exemplos

64



Exemplos

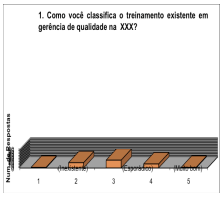
65

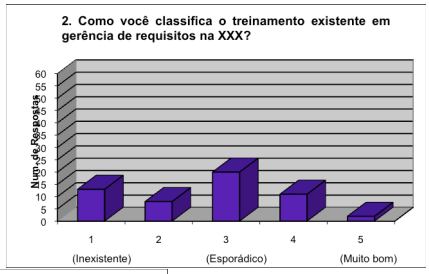
- Qualitativo
  - O quê você acha da sua formação no que se refere a produção de software de qualidade? O que você acredita ser necessário para aprimorar seu desempenho? Que conhecimentos você desejaria adquirir? Por quê?
  - Objetivo: verificar a opinião em relação a política de trainamento
  - Justificativa: uma organização madura tem políticas bem definidas de treinamento. Pergunta de controle

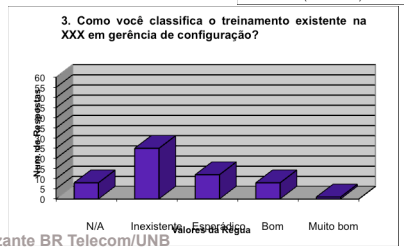
© KKB/PUC-Rio

Controle

66







© KKB/PUC-Rio

Tratamento de questionários

67

- Pacotes (SPSS)
- Análise simplificada

© KKB/PUC-Rio

Questionários

68

- +
  - padronização de perguntas
  - tratamento estatístico

© KKB/PUC-Rio

Questionários

69

- -
  - limitação das respostas
  - pouca interação/participação

© KKB/PUC-Rio

Reuniões

70

- Extensão da entrevista ou
- Participação ativa
  - períodos curtos e intensos
  - foco

© KKB/PUC-Rio

Reuniões

71

- Brainstorm
- JAD
- Workshop de Requisitos
  - participação de facilitadores
  - planejamento antecipado

© KKB/PUC-Rio

Reuniões JAD

72

Princípios do JAD

- dinâmica de grupo
- recursos visuais
- processo organizado e racional
- documentação com abordagem "o que você vê é o que você obtem" (WYSISYG)

© KKB/PUC-Rio

Reuniões JAD

73

JAD

Alvo

- identificar os requisitos de alto nível
- definir e associar a área de abrangência
- planejar as atividades da etapa de projetos
- publicar e aprovar os documentos da etapa de planos

© KKB/PUC-Rio

Reuniões - JAD

74

- Usado para **acelerar a investigação** dos requisitos do sistema, projetar a solução, definir novos procedimentos e as atividades de verificação para monitorar o projeto até a sua finalização

© KKB/PUC-Rio

Reuniões - JAD

75

- **Fator crítico** é ter todos os envolvidos relevantes presentes
- **Recomendável para projetos de 3-6 meses.** Para os mais longos, recomenda-se JAD's a cada **início de iteração.**

© KKB/PUC-Rio

Reuniões - JAD

76

- 4 princípios básicos do JAD:
  - Pessoas que **executam o negócio,** tem o melhor **entendimento do problema**
  - Pessoas capacitadas e **conhecedoras de TI** tem as melhores possibilidades de **uso da TI**

© KKB/PUC-Rio

Reuniões - JAD

77

- Sistemas de **TI** e a área de **negócio** não vivem isoladamente, há uma **correlação e inter-dependência** entre as áreas de negócio. Pessoas que atuam nestas áreas têm uma visão mais ampla e podem contribuir significativamente.
- As **melhores soluções de TI** são construídas quando estas áreas atuam juntas no projeto, como **parceiros.**

© KKB/PUC-Rio

Reuniões

78

Fases do JAD

- Customização
- Reuniões
  - apresentação de tarefas
  - junção de idéias
  - avaliação
  - compromisso
- Fechamento

© KKB/PUC-Rio

Reuniões - JAD

79

JAD - Documentos gerados

Requisitos

- requisitos de alto nível
- objetivos
- benefícios previstos
- estratégias e considerações futuras

© KKB/PUC-Rio

Reuniões - JAD

80

- **JAD - Documentos gerados**
- Requisitos
  - hipóteses e restrições
  - segurança, auditoria e controle
  - escopo do sistema
  - usuários do sistema e localização
  - áreas funcionais fora da aplicação

© KKB/PUC-Rio

Reuniões

81

JAD - Documentos gerados

Planos:

- matriz de participantes
- identificação do JAD/PROJETO
- estimativas

© KKB/PUC-Rio

Reuniões - JAD

82

- Benefícios da abordagem JAD:
  - melhora a **comunicação;**
  - **promove** o consenso e o senso de **"nosso projeto";**

© KKB/PUC-Rio

Reuniões - JAD

83

- Benefícios da abordagem JAD:
  - **Soluções inter-área** são **projetadas;**
  - aumenta a qualidade do projeto (combinação de **conhecimento = melhores soluções**)
  - Ajuda a equipe de projeto **manter o foco,** trabalhar na **"coisa certa"** no **"tempo certo".**

© KKB/PUC-Rio

Reuniões - JAD

84

- Quando Não usar o JAD:
  - Os **patrocinadores não** estão claramente **definidos**
  - **Não** há um **objetivo claro** – ausência de direção

© KKB/PUC-Rio

Reuniões - JAD

85

- Quando Não usar o JAD:
  - **Muitos integrantes** ou **muito poucos**
  - **Não há representatividade** dos envolvidos
  - **Não há comprometimento** com os **prazos**
  - Há **instabilidade** no **escopo**

© KKB/PUC-Rio

Reuniões

86

Workshop de Requisitos

- Preparação
  - local adequado
  - escolha dos participantes
  - preparar material com antecedência
  - agenda

© KKB/PUC-Rio

Reuniões

87

Workshop de Requisitos

- Facilitador
  - treinado
  - espírito de time, agregador
  - respeitado por todos os participantes
  - forte suficiente para realizar desempates

© KKB/PUC-Rio

Reuniões

88

Workshop de Requisitos

- Durante a reunião
  - descontrair os participantes
  - focalizar em idéias e não pessoas

© KKB/PUC-Rio

Reuniões

89

- **Workshop de Requisitos**
- Durante a reunião
  - abrir espaço para *brainstorming*
  - registrar o encontro
  - follow up e agenda para próximos encontros

© KKB/PUC-Rio

Reuniões

90

Workshop de Requisitos

- Problemas
  - pessoas dominadoras
  - passividade
  - atrasos
  - comentários negativos, pejorativos
  - notar que é dificil reiniciar depois de uma interrupção

© KKB/PUC-Rio

Reuniões

91

Brainstorm

- gerar e condensar idéias
- as melhores idéias são o resultado da combinação de outras
- priorizar as idéias levantadas
- ao vivo é melhor, mas podemos fazer a reunião via web ou vídeo conferência

© KKB/PUC-Rio

Reuniões

92

Brainstorm

Fase I - Geração de idéias

- Objetivo: Gerar o maior número possível de idéias
- Regras
  - não permitir críticas ou debate
  - deixar imaginação voar
  - modificar e combinar idéias

© KKB/PUC-Rio

Reuniões

93

Brainstorm

Fase II -Redução do número de idéias

- cortar as idéias que não valem a pena investir
- agrupar - criar nomes significativos e agrupar as idéias segundo estes

© KKB/PUC-Rio

Reuniões

94

Brainstorm

Fase II -Redução do número de idéias

- anotar pequenas descrições do rationale das idéias a partir de seus autores
- priorizar
  - voto
  - categorizar - crítico, importante, útilg

© KKB/PUC-Rio

Reuniões

95

- \+
  - dispor de múltiplas opiniões
  - criação coletiva

© KKB/PUC-Rio

Reuniões

96

- \-
  - dispersão
  - custo

© KKB/PUC-Rio

Observação

97

- +
  - baixo custo
  - pouca complexidade da tarefa

© KKB/PUC-Rio

Observação

98

- -
  - dependência do ator (observador)
  - superficialidade decorrente da pouca exposição ao universo de informações

© KKB/PUC-Rio

Análise de Protocolo

99

- Analisar o trabalho da pessoa “a seu lado”
- verbalização durante as tarefas
- Também pode ser utilizado por mémoria
- Objetivo - racionalização utilizada na execução de tarefas

© KKB/PUC-Rio

Análise de Protocolo

100

- +
  - fatos não observáveis
  - melhor compreensão dos fatos

© KKB/PUC-Rio

Análise de Protocolo

101

- -
  - foco na performace
  - o que se diz não é o que se faz

© KKB/PUC-Rio

Enfoque antropológico

102

- Integração ao meio ambiente
- Analista vira cliente
- Lento
- Resultados demorados
- Etnografia

© KKB/PUC-Rio

Etnografia

103 fonte: Sommerville

- Cientistas sociais passam um período considerável observando e analisando como as pessoas trabalham
- As pessoas NÃO tem de explicar ou articular como trabalham
- Aspectos sociais e organizacionais relevantes podem ser observados
- Estudos etnográficos mostram que o ambiente de trabalho é mais rico e mais complexo do que o sugerido por modelos simples.

© KKB/PUC-Rio

Etnografia Focal

104 fonte: Sommerville

- Desenvolvida durante um projeto que estudava o processo de controle de tráfego aéreo [Sommerville]
- Combina técnicas de Etnografia e dePrototipagem

© KKB/PUC-Rio

Etnografia Focal

105 fonte: Sommerville

- Desenvolvimento de protótipos resulta em questões sem resposta que vão virar o foco de estudos etnográficos.
- O problema da Etnografia é que é centrada em práticas atuais baseadas em premissas (históricas) que podem não ser mais relevantes.

© KKB/PUC-Rio

Escopo da Etnografia

106 fonte: Sommerville

- Requisitos que resultam da forma real em que as pessoas trabalham e não de como definições de processos podem sugerir
- Requisitos que resultam da cooperação e percepção das atividades de outras pessoas

© KKB/PUC-Rio

Enfoque antropológico

107

- \+
  - visão de dentro para fora
  - contextualizada

© KKB/PUC-Rio

Enfoque antropológico

108

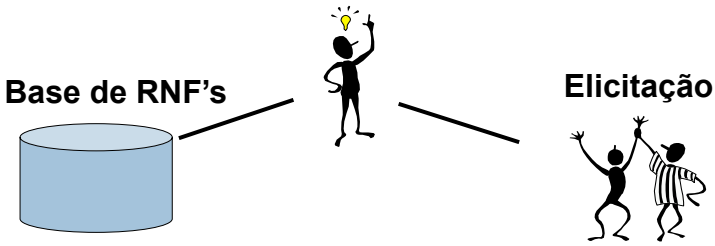
- \-
  - tempo
  - pouca sistematização

© KKB/PUC-Rio

Bases de Requisitos não-funcionais

109

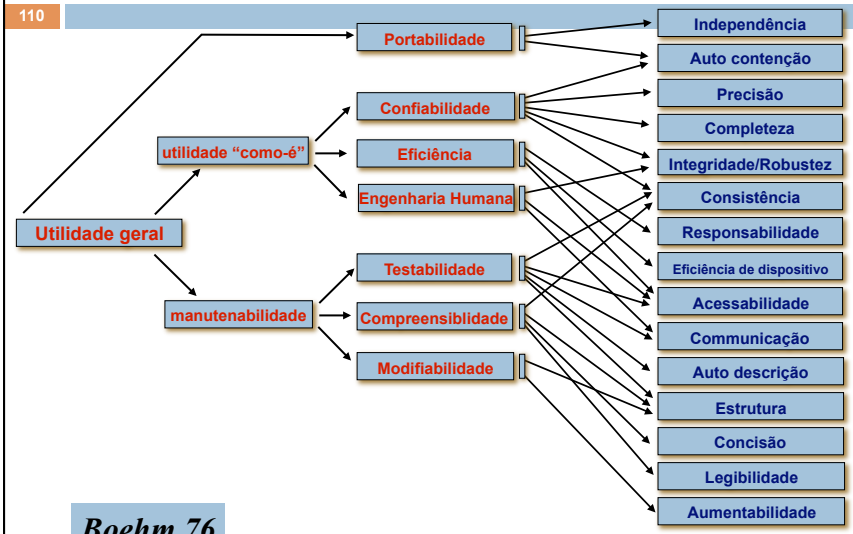
- Taxonomias propostas na literatura
- Servem de guia para a elicitação



© KKB/PUC-Rio

Taxonomia

110

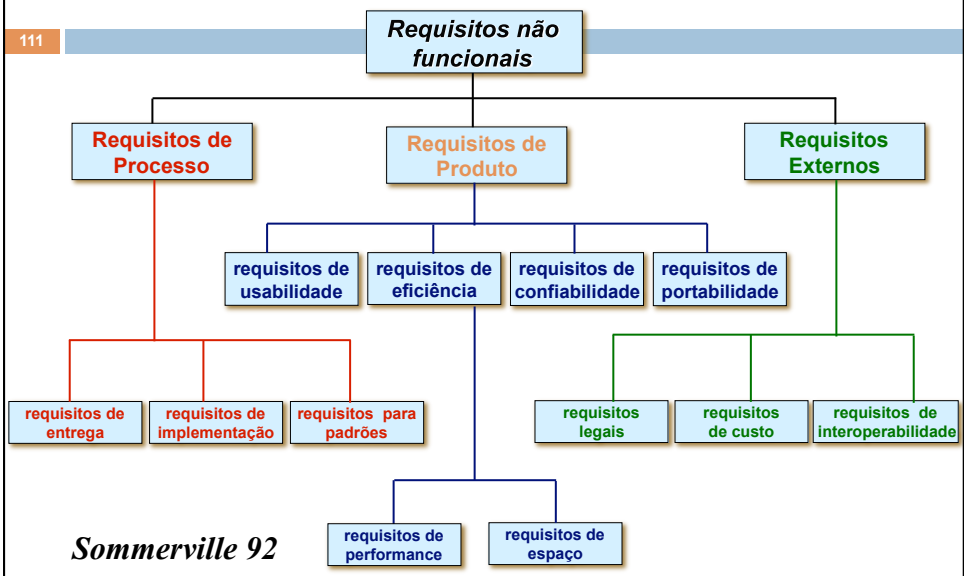


Boehm 76

© KKB/PUC-Rio

Taxonomia

111



Sommerville 92

Base de RNF's

112

- +
  - reutilização de conhecimento
  - antecipação de aspectos implementacionais
  - identifica;áo de conflitos

Base de RNF's

113

- -
  - custo de construção de base RNF
  - falsa impressão de completeza

© KKB/PUC-Rio

Participação Ativa dos Atores

114

- +
  - envolvimento de clientes e usuários
  - validação

© KKB/PUC-Rio

Participação Ativa dos Atores

115

- -
  - treinamento
  - falsa impressão de eficácia

© KKB/PUC-Rio

Engenharia Reversa

116

- +
  - disponibilidade de informação (código)
  - reuso

© KKB/PUC-Rio

Engenharia Reversa

117

- -
  - descontinuidade de informações
  - informação muito detalhada

© KKB/PUC-Rio

Reutilização

118

- Definir o que vai se reutilizar
- Definir onde se vai buscar o objeto a reutilizar
- O objeto está armazenado segundo os princípios de reutilização?

© KKB/PUC-Rio

Reutilização

119

- Dispor de mecanismos eficazes para a recuperação do componente de reutilização
- Os atores do sistema de produção estão motivados para contribuir com objetos e componentes reutilizáveis
- Seleção de ferramentas e COTS

© KKB/PUC-Rio

Seleção de Ferramentas

120

- Requisitos para seleção:
  - requisitos funcionais
  - requisitos não funcionais
  - fornecedor
  - disponibilidade

© KKB/PUC-Rio

Seleção de Ferramentas

121

Requisitos para seleção:

- metodologias suportadas
- treinamento
- integração
- custo, custo, custo

© KKB/PUC-Rio

Matriz de seleção de ferramentas

122

Avaliação

- Avaliar funções
  - gradação
    - 0,1,2 (escala ternária)
    - outras escalas - personalizadas - Ex: ruim, satisfatório, bom
    - para cada função avaliar usabilidade, integração, compreensibilidade, documentação confiabilidade

© KKB/PUC-Rio

Matriz de seleção de ferramentas

123

Avaliação

- Verficação manual ou utilização da ferramenta
- avaliar para cada pacote
  - portatilidade
  - evolução
  - plataforma
  - concordância com padrões
  - número de usuários

© KKB/PUC-Rio

Matriz de seleção de ferramentas

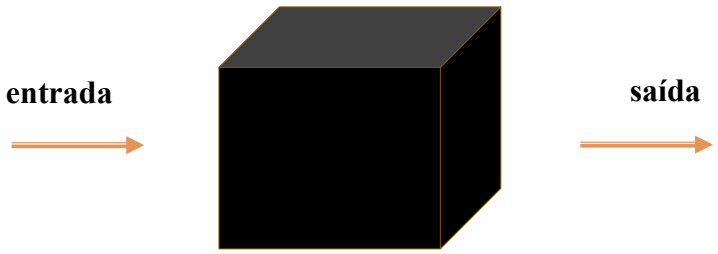
124

- gradação
  - pontuação final (1 ou mais pessoas)
  - comparar resultados
  - total da função
  - total da propriedade

© KKB/PUC-Rio

COTS

125



Caixa preta

© KKB/PUC-Rio

COTS

126

- Software pronto
- "*off the shelf*"
- Encontrado no mercado
- Ao invés de adequar o componente ao freguês escolher o software mais adequado e se adaptar a ele.

© KKB/PUC-Rio

COTS

127

- Vantagens:
  - custo
  - rapidez (disponibilidade)
  - suporte técnico
  - comunidade de usuários
  - treinamento

© KKB/PUC-Rio

Desafios

128

- Como estimar a qualidade do suporte a longo prazo?
- Como avaliar custos associados à aquisição de determinados COTS?
- Como avaliar os custos associados a manutenção de determinados COTS?

© KKB/PUC-Rio

Desafios

129

- Como verificar que o produto está em conformidade com outros padrões?
- Como descrever e qualificar os aspectos de qualidade em COTS?

© KKB/PUC-Rio

Reutilização

130

- +
  - produtividade
  - qualidade

© KKB/PUC-Rio

Reutilização

131

- -
  - nível de abstração (requisitos)
  - possibilidade de reuso real

© KKB/PUC-Rio

Qual técnica utilizar?

132

- Depende da situação
- Analisar o contexto
- Respeitar limitações

© KKB/PUC-Rio

Exemplo

133

- Área de saúde
  - Questionários
    - menos de 10% retornados
    - maioria parcialmente preenchidos
    - enfermeiras e médicos trabalham em turnos de 12 ou 24 horas

© KKB/PUC-Rio

Exemplo

134

- Área de saúde
  - Entrevistas
    - desconfiança e formalidade
    - tipo abertas tem melhor resposta
    - lista de tópicos

© KKB/PUC-Rio

Exemplo

135

- Área de saúde
  - Análise de protocolo
    - praticamente impossível (privacidade, inconveniência, tempo de resposta)
    - imagine uma emergência cardiovascular !!!!

© KKB/PUC-Rio

Exemplo

136

- Área de saúde
  - JAD
    - na maior parte para resolver conflitos
    - questões éticas
    - competitividade

© KKB/PUC-Rio

Exemplo

137

- Área de saúde
  - Vídeo e Áudio (transcritos)
    - questões legais quase impossibilitam uso
    - imagine uma emergência cardiovascular !!!!

© KKB/PUC-Rio

Exemplo

138

- Área de saúde
  - Cenários e casos de uso
    - bem aceitos
    - cenários mais utilizados (nível de detalhe)
  - Observação
    - confirmação de resultados

© KKB/PUC-Rio

139

Modelagem

© KKB/PUC-Rio

Teoria Geral de Sistemas

140

- Todo sistema é um sub-sistema de um sistema maior
- Globalidade, objetivo
- Partes interligadas visando um objetivo
- Hierarquia - estrutura arborecente
- Dividir para conquistar

© KKB/PUC-Rio

Modelagem Conceitual

- Modelo: abstração da realidade, enfatizando características específicas.
  - representar uma visão do ambiente
  - representar as partes do todo
  - permitir a abordagem gradual da complexidade (do mais abstrato para o mais detalhado)
  - úteis na organização das informações

© KKB/PUC-Rio

Modelagem Conceitual

- Em geral, um único modelo não é suficiente para representar todas as características de em sistema

© KKB/PUC-Rio

Exemplo Mapa - Endereço

143

© KKB/PUC-Rio

Exemplo Distâncias – Escala

144

© KKB/PUC-Rio

Exemplo Foto

145

© KKB/PUC-Rio

Modelagem Científica

146

- Business process modelling
- Cartography
- Climate modelling
- Data modelling
- Ecological modelling
- Economical modelling
- Environmental modelling
- Futures studies
- Geologic modelling
- Graphical modelling
- Hydrography
- Hydrological modelling
- Hydrogeological modelling
- Mathematical modelling
- Medical modelling
- Modelling in Epidemiology
- Molecular modelling
- Morphological Modelling
- Ocean modelling
- Policy modelling
- Simulation
- Software modelling
- Statistical modelling
- Stochastical modelling
- System dynamics
- Thought experiment
- Traffic and Transport modelling

Modelos Quantitativos

147

Servem para: Medir o Mundo

- Precisão (3m, 76mm, 3.896m^3, 12V, 30 minutos)
- Estatísticos
- Permitem análise automatizada

Exemplos:

Voltagem de entrada, tamanho do fêmur do bebê, tempo de cozimento, vida útil do componente)

Ref: Rector et al

© KKB/PUC-Rio

Modelos Qualitativos

148

Servem para: Descrever o Mundo:

- Pouca precisão
- Ambíguos
- Análise automatizada nos primórdios

Exemplos

Quais são os legumes saudáveis?, Melhores filmes do ano, Quais ruas tem menos trânsito?

Ref: Rector et al

© KKB/PUC-Rio

Ferramentas úteis

149

- Abstração
- Formalismo
- Decomposição

© KKB/PUC-Rio

Abstração

150

- **Ferramenta mais utilizada na racionalização de software**
- Porque?
  - Ignorar detalhes incovenientes
  - Possibilita o mesmo tratamento a entidades diferentes
  - Simplifica vários tipos de análise

© KKB/PUC-Rio

Abstração

151

- Em programação
  - Abstração é o processo de nomear objetos compostos e lidar com eles como se fossem entidades únicas
- Não resolve problemas
  - Mas simplifica!!!

© KKB/PUC-Rio

Dividir e Conquistar

152

- Tempos antigos *"divide et impera"*

DIJKSTRA (http://www.cs.utexas.edu/users/UTCS/notices/dijkstra/ewdobit.html)

(programming considered a human activity)

- Especifica as partes individualmente

© KKB/PUC-Rio

Dividir e Conquistar

153

- Satisfeito? O problema está resolvido?
- Constrói as partes individualmente

se uma das partes ainda é complexa: subdividir

© KKB/PUC-Rio

Decomposição

154

- Vantagens:
  - Pessoas diferentes podem trabalhar nos subproblemas
  - Paralelização pode ser possível
  - Manutenção é mais fácil

© KKB/PUC-Rio

Decomposição

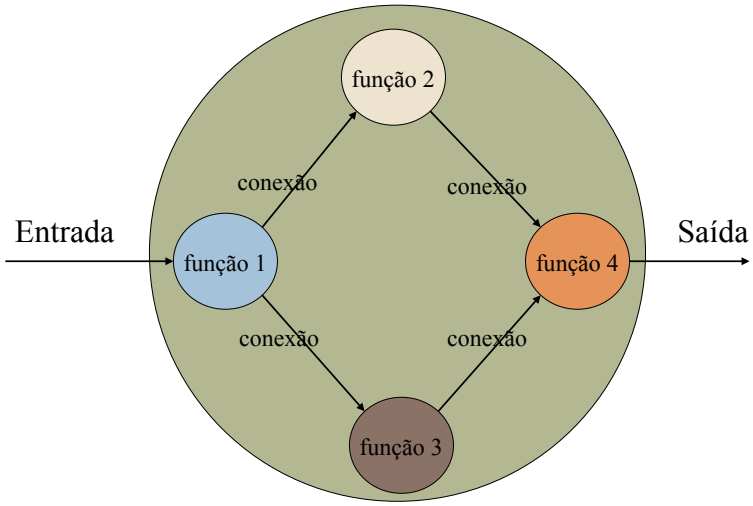
155

- Desvantagens
  - As soluções dos subproblemas podem não combinar de modo a resolver o problema original
  - Problemas de difícil compreensão são difíceis de decompor
  - A estrutura do mundo real NÃO é hierárquica [Jackson]

© KKB/PUC-Rio

Decomposição Funcional

156



© KKB/PUC-Rio

Exemplo: Taxonomia

157

- O termo taxonomia vem do grego “taxis”, que significa “arrumação”, e “nomos”, que significa “lei”.

- Definição: “vocabulário controlado e dinâmico de descritores relacionados com semântica e genericamente, que cobre de forma extensiva um ramo de conhecimento”

Dicionário Aurélio

© KKB/PUC-Rio

Exemplo

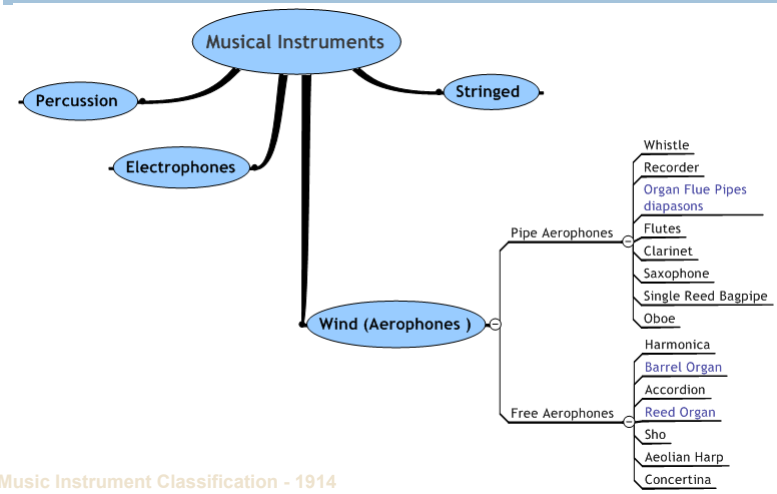
158

Kingdom: Animalia
Filo: Chordata
Subfilo: Vertebrata
Class: Mammalia
Subclass: Theria
Order: Primata
Suborder: Anthropoidea
Family: Hominidae
Genera: Homo
Species: Sapiens

© KKB/PUC-Rio

Exemplo de Taxonomia

159



Music Instrument Classification - 1914
Erich Moritz von Hornbostel & Curt Sachs

© KKB/PUC-Rio

Taxonomia "Tortas"

160

Google™ Diretório

Artes e Entretenimento
Música, Televisão, Rádios

Ciência e Meio Ambiente
Engenharia, Física, Agropecuária

Mapas e Vistas
Câmeras Online

Economia
Informática, Compras

Notícias
Revistas, Televisão, Rádio

Esportes
Futebol, Aquáticos, Artes Marciais,

Saúde
Clínicas, Hospitais

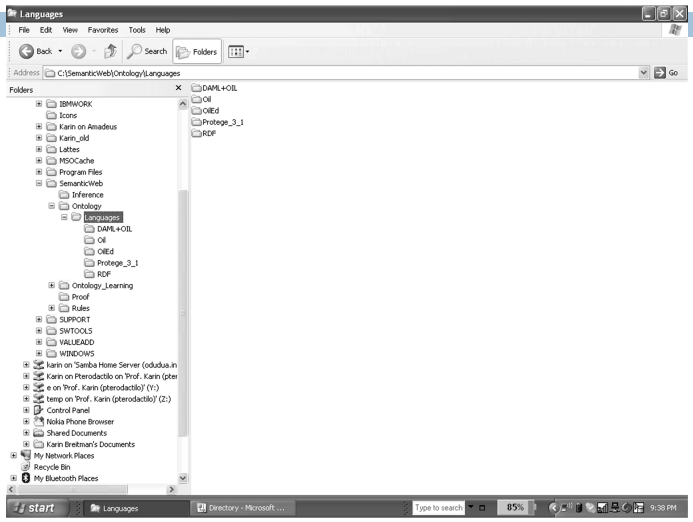
Sociedade e Cultura
Religião., Espiritualidade,

Transportes
Aéreas, Rodoviárias,

Viagens e Turismo
Minas Gerais, Hospedagem,

© KKB/PUC-Rio

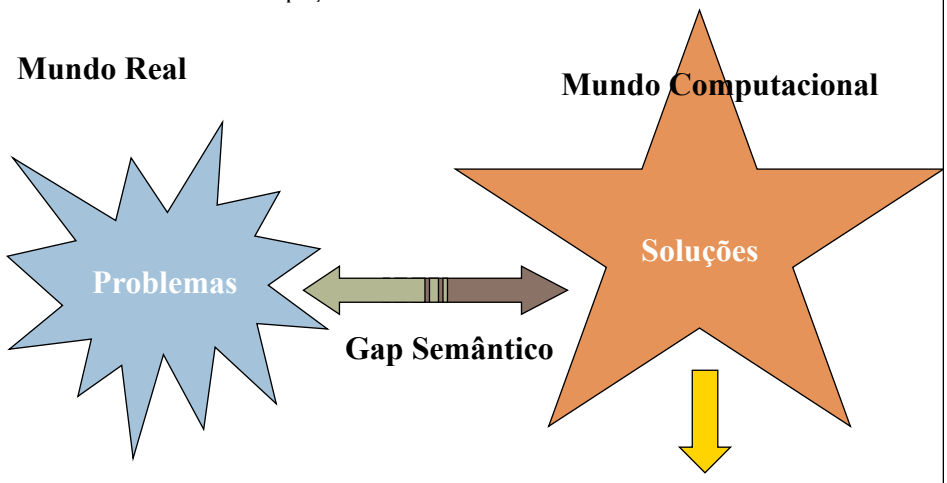
Taxonomias Tortas



© KKB/PUC-Rio

Modelagem de Sistemas

Inspiração: Guilherme Nicodemos -UCP



© KKB/PUC-Rio

Modelagem de Sistemas de Informação

- Modelos de fluxo de transformação
  - integrados a organização
  - abstraem do sistema de software
  - devem dar ênfase aos atores, aos fluxos de informação dos vários tipos

© KKB/PUC-Rio

Modelagem de Sistemas

fonte: Sommerville

- Ajuda o analista a entender a funcionalidade do sistema e a comunicação com clientes
- Modelos diferentes representam o sistema através de diversas perspectivas:
  - Extena – mostram o contexto onde o sistema é apenas parte
  - Comportamental – mostram como o sistema vai funcionar na prática
  - Estrutural – mostram a arquitetura do sistema

© KKB/PUC-Rio

Modelo

165 Fonte: Fernando Vanini - Unicamp

- Modelo: é uma abstração da realidade, enfatizando características específicas.
- Um único modelo não é suficiente para representar todas as características de em sistema

© KKB/PUC-Rio

Modelo

166 Fonte: Fernando Vanini - Unicamp

- Vários tipos de modelo no desenvolvimento de software: Análise Estrutura, ER, UML, SADT...
- Modelos são úteis na organização das informações e na especificação dos requisitos.
- Não ajudam na determinação dos requisitos.

© KKB/PUC-Rio

Histórico

Sistemas desenvolvidos sem organização

- Busca de sistemáticas de organização
- Análise Estruturada – Gane, De Marco, Yourdon -> ênfase no DFD
- Modelo Entidade-Relacionamento – Peter Chen -> ênfase no ER
- Análise Essencial – McMenamin e Palmer
- Análise Orientada a Objetos -> UML

© KKB/PUC-Rio

Dimensões Clássicas

Fonte: Fernando Vanini - Unicamp

- Um problema do mundo real pode ser visto sob perspectivas diferentes:
  - informações tratadas pelo sistema,
  - funcionalidades esperadas
  - comportamento

© KKB/PUC-Rio

Dimensões Clássicas

169

Fonte: Fernando Vanini - Unicamp

- pode-se recorrer a modelos que cobrem essas 3 dimensões relevantes ao sistema:
  - modelo de dados
  - modelo de função
  - modelo de comportamento

© KKB/PUC-Rio

Filosofias de Modelagem

170

- Sistemas
- Objetos
- Redes
- Agentes
- Metas
- Comportamento
- Situações

© KKB/PUC-Rio

Objetos

171

- Mundo é composto de objetos
- Objetos são independentes, tem memória e comportamento
- Objetos comunicam-se por mensagens
- Objetos são organizados em classes (generalizações)

© KKB/PUC-Rio

Objetos

172

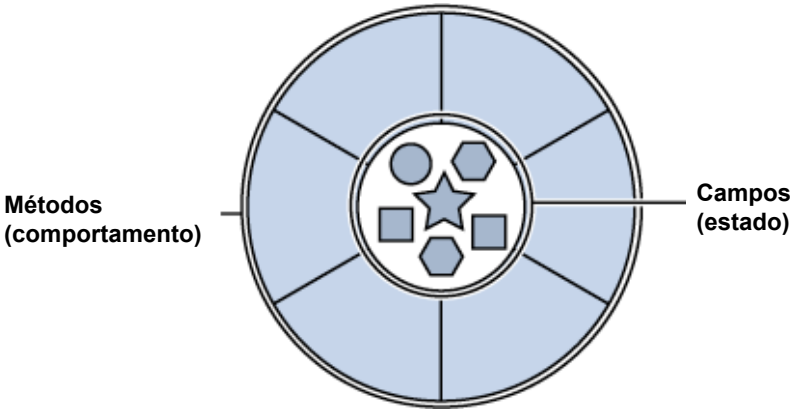
fonte:Java Sun Tutorials

- Objetos do Mundo Real: cachorro, tv, bicicleta…
- Duas características:
- Cachorro:
  - estado – nome, cor, raça, com_fome..
  - comportamento – latir, pegar, balançar rabo…
- Bicicleta:
  - estado – marcha_atual, velocidade, cadência_do_pedal
  - comportamento – trocar_marcha, mudar cadência do pedal, freiar

© KKB/PUC-Rio

Objetos

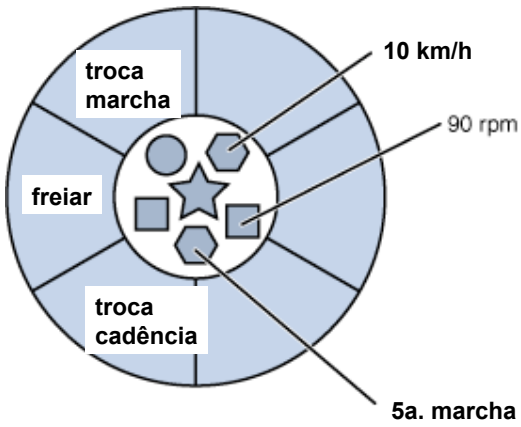
fonte:Java Sun Tutorials



© KKB/PUC-Rio

Objetos - bicicleta

fonte:Java Sun Tutorials



© KKB/PUC-Rio

Modelagem de sistemas orientada a objetos

- Propostas mais conhecidas:
  - Booch
  - OMT
  - Jacobson

© KKB/PUC-Rio

UML

Surgimento

- Diversos métodos e técnicas OO, com muitos aspectos comuns mas utilizando notações diferentes
- Dificuldades para o aprendizado, a aplicação, a construção e a utilização das ferramentas
- Diferencias entre os distintos enfoques (autores)

Necessidade de uma notação padrão

© KKB/PUC-Rio

UML

177

- "Método Unificado"
  - Grady Booch e Jim Rumbaugh
  - apresentado no OOPSLA'95
- Rational Software
  - Grady Booch, Jim Rumbaugh e Ivar Jacobson
  - Ferramenta CASE Rational Rose

© KKB/PUC-Rio

UML

178

Linguagem para modelagem e não um método

© KKB/PUC-Rio

UML

179

- Um método consiste em linguagem de notação + processo
- O processo proposto pelos autores chama-se Objectory
- Podemos utilizar UML independentemente do processo

© KKB/PUC-Rio

UML - Diagramas

180

- Diagramas de Casos de Uso
  - Diagrama de Casos de Uso
    - Atores e suas conexões com Casos de Uso
  - Descrições textuais de Casos de Uso
- Diagramas da Estrutura Estática
  - Diagrama de Classes
    - Estrutura estática das classes do sistema
  - Diagrama de Objetos
    - Exemplifica Diagrama de Classes

UML - Diagramas

181

- Diagramas de Estado
  - Diagrama de Estados
    - Estados possíveis que objetos de uma classe podem ter e eventos que causam essas mudanças de estado
  - Diagrama de Atividade
    - Fluxo seqüencial das atividades

UML - Diagramas

182

- Diagramas de Interação
  - Diagrama de Seqüência
    - Colaboração Dinâmica através "troca de mensagens" entre objetos a partir de uma função ou seqüência de tempo
  - Diagrama de Colaboração
    - Colaboração Dinâmica através da interação entre objetos (contexto)
- Diagramas de Implementação
  - Diagrama de Componentes
    - Estrutura física de código em forma de componentes de código
  - Diagrama de Distribuição
    - Arquitetura Física do Hardware e Software

Benefícios

fonte:Java Sun Tutorials

- Modularidade
  - O código de cada objeto pode ser escrito e mantido independente de outros objetos
- Esconder Informações (information hiding)
  - Detalhes de implemetação estão escondidos.
  - A interação é realizada através dos métodos – interface
- Reutilização
  - Mais fácil reutilizar código
- Manutenção
  - Mais fácil remover e verificar erros se um objeto se tornar problemático

© KKB/PUC-Rio

Desvantagens

- (Jackson):
  - a idéia de objeto é de programação e não é cabível para a maioria dos indivíduos do mundo real.
    - qual foi a última vez em que você mandou uma mensagem ao seu contra cheque?
    - quando o sol nasce ele manda uma mensagem para que os pássaros comecem a cantar?

© KKB/PUC-Rio

Objetos

"O mundo fora da máquina é muito rico, cheio de caprichos e recalcitrantemente multifacetado para ser capturado na forma de objetos"

M. Jackson

© KKB/PUC-Rio

Redes

- Comunicação entre nós interligados
- Nós representam processos independentes
- Mensagens (protocolos)

© KKB/PUC-Rio

Agentes

187

- Importância de atores e dispositivos
- Responsabilidade pela execução de processos
- Modelos orientados a agentes

© KKB/PUC-Rio

Comportamentais

188

fonte: Sommerville

- Modelos comportamentais são utilizados para descrever o comportamento do sistema como um todo

© KKB/PUC-Rio

Comportamentais

189

fonte: Sommerville

- Os dois tipos de modelos comportamentais são:
  - Modelos de processamento de Dados – mostram como os dados são processados à medida em que se fluem pelo sistema
  - Modelos de máquina de estado – mostram como o sistema responde a eventos

© KKB/PUC-Rio

Comportamentais

190

fonte: Sommerville

- Processamento de Dados x Máquina de Estado
  - Modelam perspectivas diferentes – AMBOS são necessários para descrever o comportamento do sistema

© KKB/PUC-Rio

Situações

191

- Propósito - uma situação trata da satisfação de um objetivo.
- Atores - uma situação envolve um número identificado de atores (pessoas ou aparatos ou organizações).
- Recursos - elementos que são necessários numa situação.

© KKB/PUC-Rio

Situações

192

- Tempo - representam um momento específico.
- Lugar - acontecem em um contexto geográfico.
- Restrições - podem existir pre-condições
- Concretas - são ancoradas na realidade.
- Alternativas - podem levar a ações alternativas.

© KKB/PUC-Rio

Regras de Negócio

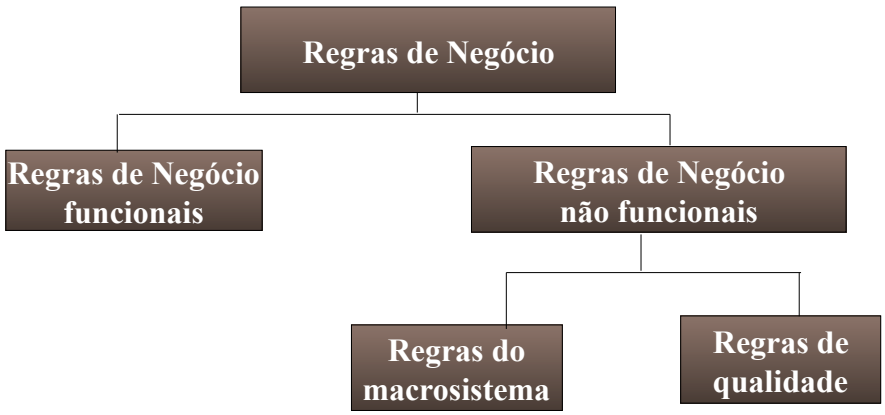
193

- Políticas (não procedimentos)
- identificação
  - limites
  - responsabilidades
  - direitos
- Estabilidade

© KKB/PUC-Rio

Regras de Negócio

194



© KKB/PUC-Rio

Exemplos

Regras funcionais

- Uma reunião pode ser replanejada ou cancelada
- Os supervisores respondem ao diretor geral

© KKB/PUC-Rio

Exemplos

Regras Macrosistema

- O imposto de renda deve ser recolhido mensalmente

© KKB/PUC-Rio

Modelagem Orientadas à Especificação

197

- Durante algum tempo vistas como técnicas de requisitos (análise)
- Várias técnicas: das mais formais as mais utilizadas por profissionais.

© KKB/PUC-Rio

Modelagem Orientadas à Especificação

198

Algumas delas (mais utilizadas):

- DFD, JSD, Tabelas de Decisão, Máquinas de Estado (StateChart), SADT, Eventos Externos, MER, Dicionário de Dados.

© KKB/PUC-Rio

Análise Estruturada

199

- "Utilização de Ferramentas que permitem a especificação formal dos requisitos do sistema a ser desenvolvido".
  - Estudo de um problema que antecede à tomada de uma ação ...
  - Estudo de uma área de trabalho que leva a especificação de um sistema informatizado.

© KKB/PUC-Rio

Atores

200

- Usuário – Conhece o problema mas não sabe como resolve-lo

computacionalmente ...

- Analista – Traduz as necessidades do usuário em especificações formais-técnicas necessárias aos programadores ...
- Programador – Capazes de resolver o problema caso o compreendam ...
- Gerente de projeto
- Analista de Sistemas

© KKB/PUC-Rio

Dificuldades

201 Fonte: Luis Garcia

- Dificuldade de aprendizagem e contextualização da organização sendo analisada

"Construímos um sistema tecnicamente excelente, mas não era o que os usuários queriam ...."

© KKB/PUC-Rio

Dificuldades

202 Fonte: Luis Garcia

- Comunicação – Envolvidos no processo com experiências, conhecimentos, preconceitos e terminologias diferentes ...

"Sei que você acredita que entendeu o que acha que eu disse, mas não estou certo de que percebe que aquilo que ouviu não é o que eu pretendia dizer ...".

© KKB/PUC-Rio

Dificuldades

203 Fonte: Luis Garcia

- Resistência a mudanças ...
- Problemas com estudos de viabilidade e definições claras por parte dos usuários ...
- Investimento (tempo e dinheiro)

© KKB/PUC-Rio

Dificuldades

204 Fonte: Luis Garcia

- Volatilidade dos requisitos
- Sobrecarga de detalhes
  "Distinguir a floresta das árvores ...."
- Conhecimento Tácito
  "Saber o que o usuário pensa mas não diz ..."
- Preferências dos usuários – intuitivas ...
- Documentos de projeto para usuários X técnicos

© KKB/PUC-Rio

Objetivos

205

- Particionamento da aplicação
  - Diminuição de riscos
  - Paralelismo
- Modelo Lógico da aplicação
- Permitir Visualização do Sistema como um todo
- Construir sistemas fáceis de manter

© KKB/PUC-Rio

Modelos da AE

206

- DFD – Diagrama de Fluxo de Dados
- DER – Diagrama de Entidade e Relacionamento
- DTE – Diagrama de Transições de Estado
- DD – Dicionário de Dados

© KKB/PUC-Rio

Data Flow Diagrams

207 fonte: Ian Sommerville

- DFDs modelam o sistema sob uma perspectiva FUNCIONAL
- Rastrear e documentar os dados associados a um processo ajuda no entendimento do sistema
- DFDs podem ser utilizados para mostrar a troca de informação entre sistemas diferentes

© KKB/PUC-Rio

Levantamento dos requisitos

208 Fonte: Luis Garcia

- Identificar as pessoas "Certas"
- Evitar disperdício de tempo e desgaste do projeto
- Basear-se no organograma da empresa
- Definir 3 níveis: Operativo, Superior Operativo e Gerencial
- Definir seqüência de acordo com o fluxo organizacional da empresa

© KKB/PUC-Rio

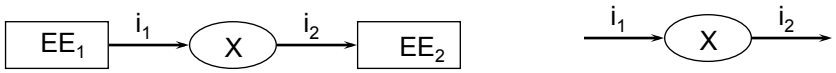
Levantamento dos requisitos

209 Fonte: Luis Garcia

- Planejar a Entrevista
- Definir Objetivos
- Determinar o roteiro, mas permitir um certo grau de improvisação
- Lista de perguntas entregues com antecedência
- Validar as entrevistas
- Utilizar ferramentas automatizadas (gráficas, de prototipação)

© KKB/PUC-Rio

DFD



→ processo (tarefa, ação, atividade) X

→ depósito de dados y

© KKB/PUC-Rio

DFD

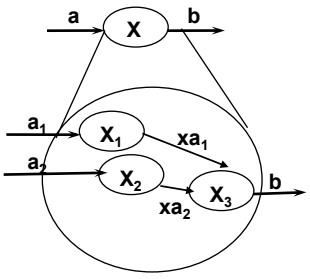
→ entidades externas (origem/destino) EE_1

→ fluxos de dados i →

© KKB/PUC-Rio

DFD

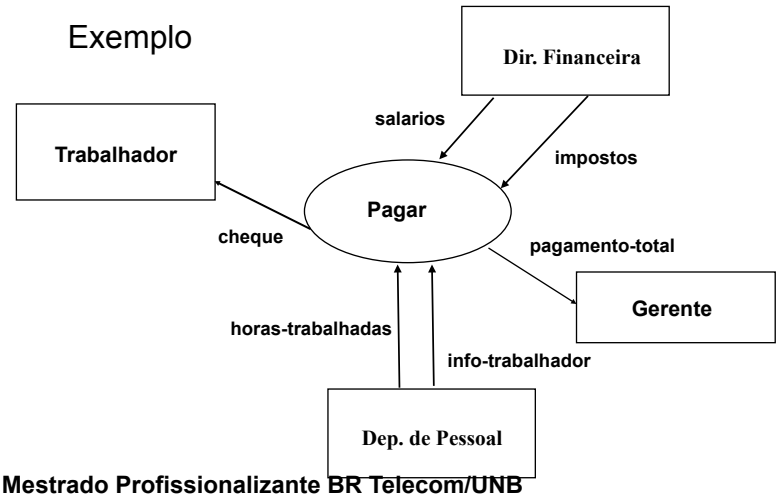
-Decomposição funcional
- -diagrama de Contexto
- - nível 0, 1, 2, …
- - tudo que entra entra, tudo que sai, sai
- - [3,6]
- - condição de parada de decomposição (quando poder descrever em uma página com pseudo linguagem)



© KKB/PUC-Rio

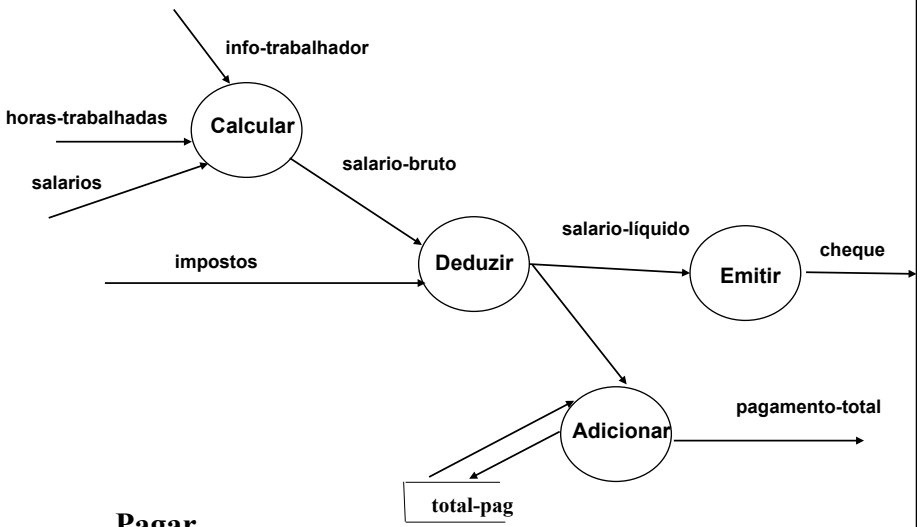
DFD

Exemplo



© KKB/PUC-Rio

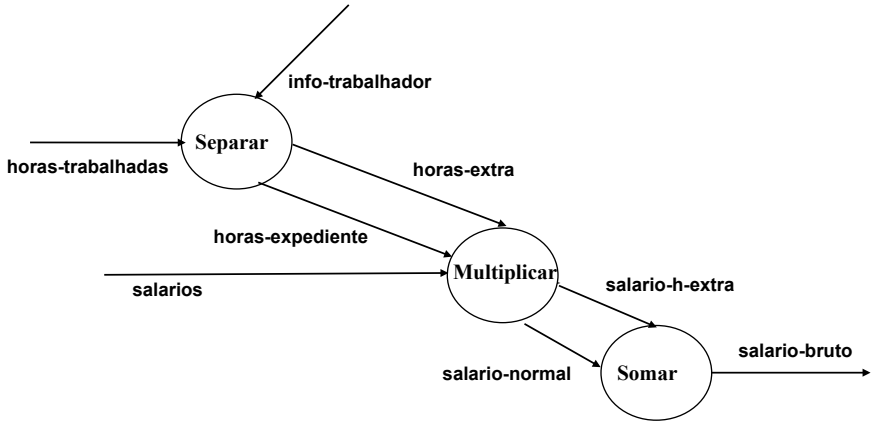
DFD



Pagar

© KKB/PUC-Rio

DFD



Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio

DFD

- Regras:
  - Nomes:
    - Processo: verbo ou frase verbal
    - Fluxo: substantivo ou frase substantiva (uso de hifens)
    - Depósito: frase substantiva ou substantivo
    - Entidate Externa: substantivo ou frase substantiva no papel de sujeito
    - Nomes diferentes (entrada é sempre diferente de saída)
    -Processo, Entidade Externa e Depósito: letra maiúscula
    - Fluxos: letra minúscula
  - Grafo:
    -Não existe entrada sem processo e sem saída
    - Fluxos para depósitos não precisam ser nomeados

Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio

Modelos de Máquina de Estado

- Modelam o comportamento do sistema em relação a eventos internos e externos
- Mostram a resposta do sistema a estímulos e por isto são muito utilizados em modelam de sistemas de tempo real
- Mostram estados como nós e eventos como arcos entre estes nós. Quando um evento ocorre, o sistema muda de um estado para outro.
- O modelo de Statecharts é parte da notação UML e serve para representar modelos do tipo máquina de estado.

© KKB/PUC-Rio

Máquinas de Estado

- **Notação**

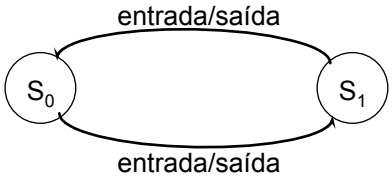
Estados são estados do sistema visíveis externamente.

(Exemplos:
Elevador no 5 andar
Interruptor ligado
Número da conta é incorreto)

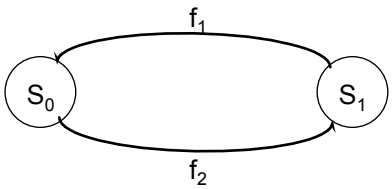
© KKB/PUC-Rio

Máquinas de Estado

• Notação



"orientado a dados"



"orientado a função"

© KKB/PUC-Rio

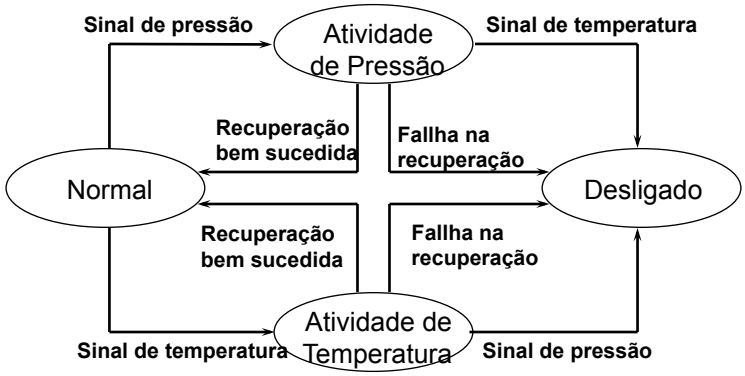
Máquinas de Estado

- Suponha o controle (parcial) de uma fábrica de produtos químicos. Ghezzi et al
  - • temperatura e nível de pressão têm que ser monitorados por questões de segurança.
  - • sensores são instalados para gerar sinais apropriados quando qualquer desses níveis excedem determinado limite.
  - • avisos são emitidos se existe um superaquecimento ou se os valures são muito altos.generate
  - • quando um desses sinais é ativado, um atividade de recuperação de temperatura ou pressão é executada.
  - • se a atividade de recuperação não é bem sucedida, então o sistema desliga a fábrica.

© KKB/PUC-Rio

Máquinas de Estado

- Exemplo:



© KKB/PUC-Rio

Máquinas de Estado

- Vantagens
  - fáceis de aprender e executar
  - modelo importante para descrever aspectos de controle
  - utilizado em diferentes situações para descrever comportamento de controle de fluxo (GUIs, protocolos, sistemas de comunicação)

© KKB/PUC-Rio

Máquinas de Estado

- Desvantagens

-explosão de estados

-modelo síncrono

© KKB/PUC-Rio

StateCharts

- **Estrutura**

  Decomposição hierárquica em componentes físicos (módulos), com os fluxos de informação entre eles.

- **Funcionalidade**

  A funcionalidade do sistema é decomposta e descrita pelas atividades (activity-charts). Os fluxos de dados e de controle entre as atividades são fornecidos.

© KKB/PUC-Rio

StateCharts

- **Comportamento**

O comportamento é fornecido através da descrição das atividades de controle, em cada nível de decomposição. Essa descrição engloba: as atividades, os fluxos de dados e o fluxo de controle.

© KKB/PUC-Rio

StateCharts

- StateCharts são essencialmente maquinas de estado, com a adição de:

  • estados hierarquicos e ortogonais (evitando a explosão de estados)

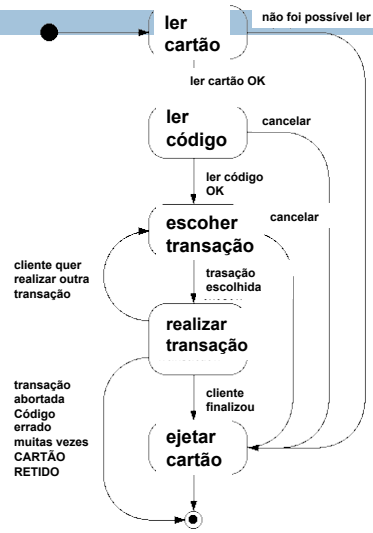
  • dados

  • tempo (temporizadores discretos)

  • difusão de sinais de controle (permitindo sincronicidade)

© KKB/PUC-Rio

Caixa Eletrônico



© KKB/PUC-Rio

Diagramas de Transição de Estado

- Vantagens
  - fáceis de aprender e executar
  - modelo importante para descrever aspectos de controle
  - utilizado em diferentes situações para descrever comportamento de controle de fluxo (GUIs, protocolos, sistemas de comunicação)

© KKB/PUC-Rio

Diagramas de Transição de Estado

- Desvantagens

-explosão de estados
-modelo síncrono

© KKB/PUC-Rio

Modelo Entidade Relacionamento

- Objetivo

  Fornecer uma descrição dos dados e de seus relacionamentos. É utilizado para descrever o modelo conceitual de bancos de dados.

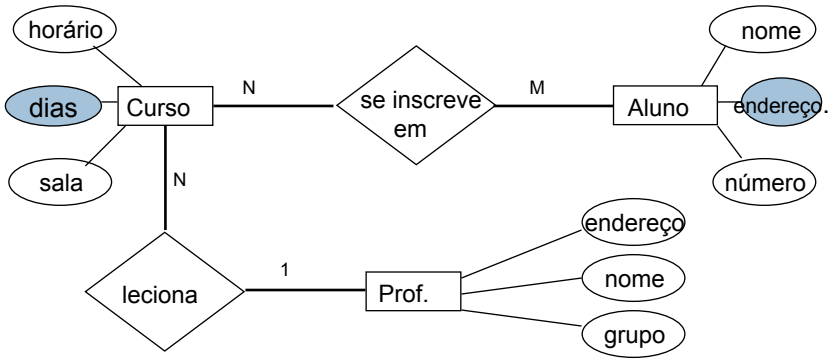
- Conceitos

  - **Entitidades** [tipo]
  - **Relacionamentos**
    → cardinalidade (1:1, 1:M, M:N)
  - **Atributos** (caracterizam entidades ou relacionamentos)

© KKB/PUC-Rio

Modelo Entidade Relacionamento

- Exemplo



© KKB/PUC-Rio

Modelo Entidade Relacionamento

- Vantagens

- notação conhecida

- mapeia os dados (data map)

- fácil de transformar em no esquema físico (relacional)

© KKB/PUC-Rio

Modelo Entidade Relacionamento

- Desvantagens
  - **dificuldade em representar restrições**
  - **não representa comportamento**
  - **semântica fraca (se baseia em hipóteses e linguagem natural)**

© KKB/PUC-Rio

Modelagem de Requisitos

- Modelos (linguagens para construção de sistemas de software)
- Modelos de requisitos
- Modelos de especificação
- Modelos de múltiplos propósitos

© KKB/PUC-Rio

Elicitação

235

Perguntar *porquê?*

"A cafeteira deve ser feita de aço"

- qual a razão disto?
- pode me explicar porquê?
- qual o pensamento atrás disto?

© KKB/PUC-Rio

Elicitação

236

"Porque se for de vidro pode quebrar"

Requisito real

- A cafeteira deve ser feita de material inquebrável
  - Plástico
  - Poliuretano
  - Até mesmo aço

© KKB/PUC-Rio

Exercício

237

- "a cafeteira tem uma luz vermelha que pisca quando está ligada, quando a água chega na temperatura certa ela fica ligada (sem piscar)"
  - Quais seriam as perguntas do tipo "porque" que poderiam ser utilizadas nesta situação?
  - Quais seriam os "reais" requisitos?

Dica: Separar requisito de solução/implementação

© KKB/PUC-Rio

Observação

238

- Os usuários misturam a solução com os requisitos

Separar NECESSIDADE da solução proposta ou atual!

© KKB/PUC-Rio

Sentenças de Requisitos

239

- O sistema deve + [verbo + objeto | frase verbal] + [complemento de agente | nulo] + [condições | nulo]
- Três classes de sentenças: {Entrada, Saída, Mudança de Estado}
- Verbo é um verbo simples que expresse a funcionalidade daquele requisito
- Frase verbal é uma frase que expressa a funcionalidade do requisito
- Complemento de agente é a identificação de um agente relacionado com o requisito. Algumas vezes esse complemento pode ser descrito pelo objeto indireto. Um agente pode ser uma pessoa, uma instituição, um grupo ou um dispositivo físico externo ao software
- Como vimos as sentenças podem ser de três tipos: funcionais, não-funcionias e inversas.

© KKB/PUC-Rio

Sentenças de Requisitos

240

- O sistema deve emitir um recibo para o cliente.
- O sistema deve emitir o recibo para o cliente em no máximo 2 segundos.
- O sistema deve cadastrar o cliente, desde de que o cliente possua identificação.
- O sistema deve transformar uma fita disponível em fita emprestada, quando a fita for alugada pelo cliente.
- O sistema deve cadastrar bibliotecários.
- O sistema deve verificar a identidade do bibliotecário.
- O sistema não pode deixar que um livro fique na reserva mais de um mês.
- O sistema deve tornar um livro em livro emprestado, quando um usuário pegar este livro emprestado.

© KKB/PUC-Rio

Clareza

Um requisito claro	
Tipo de usuário	O engenheiro de teste…
Resultado desejado	…simula…
Objeto	…erros de componente ….
Condições	…utilizando as funções de teste QQ e TT.
Um requisito vago	
	Em geral o sistema…
Precisa ou não?	… deve ser capaz…
Quais?	…de diagnosticar possíveis erros…
Como verificar isto?	… em um prazo razoável.

Ambiguidade

- "O sistema deve enviar relatórios de produtividade dos programadores, analistas ou desenvolvedores do projeto mensalmente ou quando requisitado."
- "Realizar rotina de importação de dados periódica de preço de fluido"
- "Identificar e associar as intervenções que são complementares às outras"
- O sistema deve emitir uma mensagem de atenção visual ou auditiva no evento de falha do sistema de refrigeração.

© KKB/PUC-Rio

Requisitos Incompletos

243

- Curva S (Planejado X Realizado) de um projeto
- Cadastro de iniciativas estratégicas
- Cadastro de iniciativas de melhoria
- Acompanhamento das atividades
- Acompanhamento dos projetos (percentual de conclusão)

© KKB/PUC-Rio

Requisitos Múltiplos

244

- No evento de falha da rede elétrica, o sistema deve enviar mensagem de erro ao usuário, salvar a configuração atual do sistema e os dados entrados, até então.
- O sistema deve manter dados estatísticos sobre compra, venda e movimentação do estoque de materiais de limpeza. Informação relativa a comissão de vendedores também deve ser mantida.
- Cadastro das atividades de um projeto e produtos e funcionário alocados na atividade

© KKB/PUC-Rio

Requisitos com alternativas

Mas, com exceção, apesar, quando…

- O sistema deve mostrar o total do pedido a medida em que os códigos dos produtos vão sendo entrados no pedido, a não ser que se trate de um produto promocional.

© KKB/PUC-Rio

Requisitos mal escritos

- (Projetos coordenados por um funcionário)
- Atividades responsáveis por um funcionário
- O sistema poderá ser acessado remotamente por qualquer unidade internacional da Petrobras, com isso, ele deverá ter um desempenho compatível ao acesso.
- Na improvável eventualidade de falha no sistema de refrigeração, o sistema deve mandar mensagem para a chave *admin*

© KKB/PUC-Rio

Palavras a evitar

247

- Ou – ambiguidade
- E, ou , com , também – múltiplos requisitos
- Mas, com exceção de, a não ser, apesar, sempre – requisitos com alternativas

© KKB/PUC-Rio

Exercício:

248

- Estes requisitos iniciais contém alguns defeitos que podem se tornar problemas mais adiante no projeto. Identifique estes defeitos e refraseie os requisitos de acordo.
- “O motorista deve obter resposta imediata ao fornecer qualquer comando de voz”
- "O sistema deverá estar acessível na sede da Petrobrás e nas diversas áreas"
- “Se o usuário fornecer informações erradas o sistema deve sinalizar através de mensagem de erro e indicação na tela.”
- “O sistema deve fornecer confirmação visual da ação de envio de dados, com a exceção do formulário xyz”
- “O mesmo subsistema deve ser responsável pelo envio mensal do relatório de atividades”
- "Necessidade de padronização de todas as planilhas "

© KKB/PUC-Rio

Elicitação

249 Documentos do domínio

- a) o projeto é de um banco de dados que mantém o histórico de desempenho dos projetos.b) vai permitir que usuários acessem cronogramas originais e c) compará-los com projetos atuais. d) O sistema deve ser fácil de usar pelos gerentes e e) acessível de qualquer máquina na intranet. f) O sistema deve ser implementado em Oracle e g) ter boa performance. h) o sistema deve produzir um relatório de progresso de todos os projetos para o gerente geral mensalmente.
- i) a tela deve mostrar as tendências das datas previstas versus marcos atingidos e j) prever as datas prováveis de término dos projetos.
- k) O sistema deve estar implementado em Dezembro de 2004 e l) será desenvolvido pela Informática e m) e controlado pela Gerência de Informação. n) O software deve rodar em PCs.

© KKB/PUC-Rio

Elicitação

250

- Para as frases a – n classificar em:

- Requisitos do usuário:
- Requisitos do sistema:
- Elementos de projeto:
- Planejamento:
- Informação:

© KKB/PUC-Rio

Requisitos não funcionais

- Algumas palavras levam a requisitos impossíveis de serem verficados

Palavras não Verificáveis	Possíveis substitutos
Amigável	Número máximo de passos Lista de funcionalidades presentes em outras aplicações Menus ou prompts para auxiliar usuários
Portável	Dimensões Requisitos mínimos de hardware Sistemas operacionais em que deve funcionar
Pequeno	Dimensões aceitáveis (em número de Bytes)
Flexível	Variáveis que podem acomodar uma gama de mudanças de valores Funções que implementam uma de várias possibilidades

Ralph Young – effective requirements practices

Exemplo

Requisitos não funcionais devem ser elaborados até que se tornem verificáveis

Requisito Inverificável	Requisito Verificável
“ O banco de dados ZZ deve ser flexível”	•O banco de dados ZZ deve possuir oito campos por registro.
“MNOP deve ser seguro”	•MNOP deve parar sua operação se uma pessoa se aproximar a mais de 2 metros de uma de suas partes móveis •MNOP deve desligar os aquecedores se a temperatura da água exceder 37°C •MNOP deve estar dentro dos padrões estabelecidos pela norma N567 seção 3.6 para temperaturas de superfícies externas.
“O sistema CE deve processar depósitos rapidamente”	•O sistema CE deve escanear os dados do usuário e conta de cada folheto de depósito em 2 segundos ou menos”

Ralph Young – effective requirements practices

Exercício

253

- Elabore os requisitos não funcionais abaixo para um sistema de compra de livros pela Internet
  - O sistema deve prover segurança dos dados de pagamento de seus clientes.
  - O sistema deve prover facilidades de busca de produtos de seu catálogo.

© KKB/PUC-Rio

Documento de Requisitos: qualidade

254

- critérios de qualidade para o documento de requisitos
- buscando a qualidade por diferentes meios:
  - tratamento de riscos
  - uso de métricas
  - técnicas para identificação antecipada de problemas

© KKB/PUC-Rio

Documento de Requisitos: qualidade

255

Necessidade	**O sistema é capaz de atingir seus objetivos sem este requisito? Caso afirmativo este é um requisito desnecessário**
Verificável	**É possível verificar que este requisito está sendo atendido pelo sistema?**
Atingível	**O requisito pode ser atendido pelo sistema que está sendo desenvolvido?**
Livre de Ambiguidades	**O requisito possui mais de uma interpretação possível?**
Rastreável	**A origem dos requisitos é conhecida? O requisito pode ser referenciado e localizado no sistema?**
Alocação	**O requisito pode ser alocado a um elemento ou componente do sistema?**
Concisão	**O requisito está descrito de forma simples e concisa?**
Livre de Implementação	**O requisito descreve o QUE deve ser feito sem influências de possíveis implementações?**
Correção	**O requisito contém toda a informação necessária que permita sua implementação?**
Priorizável	**O requisito é passível de ser priorizado frente aos outros requisitos?**

© KKB/PUC-Rio

Documento de Requisitos: qualidade

256

- características adicionais desejáveis em Requisitos agregados:
  - ser completo
  - ser consistente no conjunto

© KKB/PUC-Rio

Documento de Requisitos: qualidade

257

- características de suporte:
  - identificação unívoca
  - classificação (ex: funcional, não funcional, inverso)

© KKB/PUC-Rio

Documento de Requisitos: riscos

258

- maiores riscos de um DR:
  - "passar por cima" de um requisito crucial
  - definir requisitos incorretamente
  - representar inadequadamente os clientes

© KKB/PUC-Rio

Documento de Requisitos: riscos

259

- maiores riscos de um DR:
  - modelar apenas aspectos funcionais
  - falta de inspeções nos requisitos
  - buscar a perfeição nos requisitos antes de iniciar a construção do software

© KKB/PUC-Rio

Documento de Requisitos: riscos

260

"PASSAR POR CIMA" DE UM REQUISITO CRUCIAL

- acontece quando requisitos intrínsecos ao domínio não são expressos pelo cliente (ele pressupõe que já há conhecimento) e não são conhecidos dos desenvolvedores
- também ocorre com requisitos não funcionais (escalabilidade, segurança, desempenho, ...)

© KKB/PUC-Rio

Documento de Requisitos: riscos

261

PASSAR POR CIMA" DE UM REQUISITO CRUCIAL

Recomendação:

- o domínio e o vocabulário específico devem ser bem compreendidos antes que os requisitos sejam expressos pelos desenvolvedores (por exemplo, com o uso de glossários, léxicos ou ontologias)

© KKB/PUC-Rio

Documento de Requisitos: riscos

262

DEFINIR REQUISITOS INCORRETAMENTE

- se o requisito está expresso associado ao *design* (estrutura), então temos uma solução, não um requisito (independência da implementação)

© KKB/PUC-Rio

Documento de Requisitos: riscos

263

DEFINIR REQUISITOS INCORRETAMENTE

- por outro lado..... se o sistema é destinado a operar em hardware desenvolvido especialmente para ele, porque não representar o requisito como desenho???

Recomendação:

- evitar definir requisitos associados à implementação

© KKB/PUC-Rio

Documento de Requisitos: riscos

264

REPRESENTAR INADEQUADAMENTE OS CLIENTES

- uma categoria de usuário não representada no processo de requisitos poderá gerar solicitações que exijam grandes alterações em componentes, ou mesmo mostrar a necessidade de mudanças na própria estrutura do software

© KKB/PUC-Rio

Documento de Requisitos: riscos

REPRESENTAR INADEQUADAMENTE OS CLIENTES

Recomendação:

- interação com clientes e usuários é fundamental na negociação e priorização de requisitos
- clientes/usuários devem validar o que foi acordado

© KKB/PUC-Rio

Documento de Requisitos: riscos

MODELAR APENAS ASPECTOS FUNCIONAIS

- o processo de elicitação enfatiza aspectos funcionais
- atributos como integridade, desempenho, segurança, robustez, facilidade de uso, escalabilidade, inovação estão no centro da questão relativa à valorização do software pelo usuário

© KKB/PUC-Rio

Documento de Requisitos: riscos

267

MODELAR APENAS ASPECTOS FUNCIONAIS

- *casos de uso, cenários e user stories* – refletem situações onde o sistema deverá atuar, enfatizando funcionalidades, e podem não considerar atributos dos requisitos

Recomendação:

- utilizar taxonomias de RNF´s e complementar o conjunto de requisitos elicitados

© KKB/PUC-Rio

Documento de Requisitos: riscos

268

FALTA DE INSPEÇÕES EM REQUISITOS

- custo da remoção de defeitos em requisitos cresce geometricamente com o tempo
- "não foi examinado conjunto de requisitos sem defeitos" [Linscomb03]

© KKB/PUC-Rio

Documento de Requisitos: riscos

269

FALTA DE INSPEÇÕES EM REQUISITOS

Recomendação:

- executar inspeções com equipes internas ou externas à organização
- inspeções detectam diversos tipos de problemas; até 80% dos defeitos podem ser encontrados e removidos
- retorno sobre investimento: da ordem de 10 para 1

© KKB/PUC-Rio

Documento de Requisitos: riscos

270

BUSCA DA PERFEIÇÃO NOS REQUISITOS

- requisitos evoluem ao longo do processo de desenvolvimento (para projetos com razoável estabilidade, taxa média de 1% ao mês)
- desenvolvimento do projeto mostra a usuários possibilidades não conhecidas; desenvolvedores aprendem mais sobre o domínio da aplicação

© KKB/PUC-Rio

Documento de Requisitos: riscos

271

BUSCA DA PERFEIÇÃO NOS REQUISITOS

- princípio da incerteza de Heisenberg para requisitos: “assim que alguma coisa é colocada no papel, ela muda” ...
- alterações fazem parte do processo; se for o caso, modificações serão incorporadas à versão seguinte

© KKB/PUC-Rio

Documento de Requisitos: riscos

272

BUSCA DA PERFEIÇÃO NOS REQUISITOS

Recomendação:

- o efetivo gerenciamento de requisitos, com uso da rastreabilidade, auxilia nas previsões de alteração em cronogramas e custos, quando uma mudança é solicitada; esses itens serão utilizados na negociação entre equipe de desenvolvimento e cliente

© KKB/PUC-Rio

Documento de Requisitos: métricas

273

- Como as métricas podem auxiliar a verificação da qualidade
  - no processo de requisitos?
  - no documento de requisitos gerado?
- métricas devem ser utilizadas e armazenadas numa *baseline* para permitir análise da evolução do projeto, e servir como comparação para outros projetos

© KKB/PUC-Rio

Documento de Requisitos: métricas

274

- algumas métricas simples para permitir analisar a evolução dos trabalhos:
  - estabilidade de requisitos: o número de novos requisitos se mantém estável ao longo do tempo
  - volatilidade de requisitos: número de requisitos alterados/excluídos ao longo do tempo

© KKB/PUC-Rio

Documento de Requisitos: métricas

275

- algumas métricas simples para permitir analisar a evolução dos trabalhos:
  - taxa de defeitos: percentual de requisitos registrados incorretamente
  - taxa de validação: percentual de requisitos já validados pelos usuários
  - taxas de rastreabilidade: indicativas das ligações de requisitos às suas origens e aos artefatos gerados nas fases de design e implementação, como componentes e casos de testes

© KKB/PUC-Rio

Documento de Requisitos: métricas

276

- estabilidade em requisitos:

$$\text{tax\_est:} \quad \frac{n_{dr}}{n_r}$$

nrq número de novos requisitos/período

nr número total de requisitos

- durante o processo de requisitos, espera-se que as taxas de novos requisitos por período sejam altas, pois os requisitos ainda estão sendo definidos junto aos usuários. Após o processo de requisitos, espera-se uma taxa próxima de 1% ao mês

© KKB/PUC-Rio

Documento de Requisitos: métricas

277

- volatilidade dos requisitos:

$$\text{tax\_alt:} \quad \frac{n_{ra}}{n_r}$$

nra número de requisitos alterados e/ou excluídos

nr número total de requisitos

- alterações em requisitos já validados devem ser rigorosamente controladas; geralmente solicitações de mudança são avaliadas e negociadas junto aos clientes e usuários.

© KKB/PUC-Rio

Documento de Requisitos: métricas

278

- taxa de defeitos em requisitos:

$$\text{tax\_def:} \quad \frac{n_{dr}}{n_r}$$

ndr número de defeitos em requisitos

nr número total de requisitos

- defeitos em requisitos são identificados através de inspeções; defeitos não comprometem a funcionalidade do software, dado que são detectados antes da implementação estar finalizada.

© KKB/PUC-Rio

Documento de Requisitos: métricas

279

- taxa de validação de requisitos:

$$\text{tax\_val:} \quad \frac{n_{rv}}{n_r}$$

nrv número de requisitos validados sem problemas

nr número total de requisitos

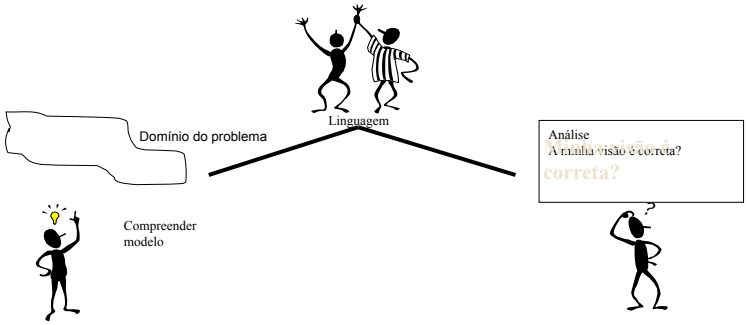
- o processo de validação envolve clientes e usuários do sistema sendo construído; são estas as pessoas que têm o problema a ser resolvido ou atividade a ser automatizada

© KKB/PUC-Rio

Léxicos e Glossários

280

- Necessidade de garantir boa comunicação entre clientes/usuários e desenvolvedores

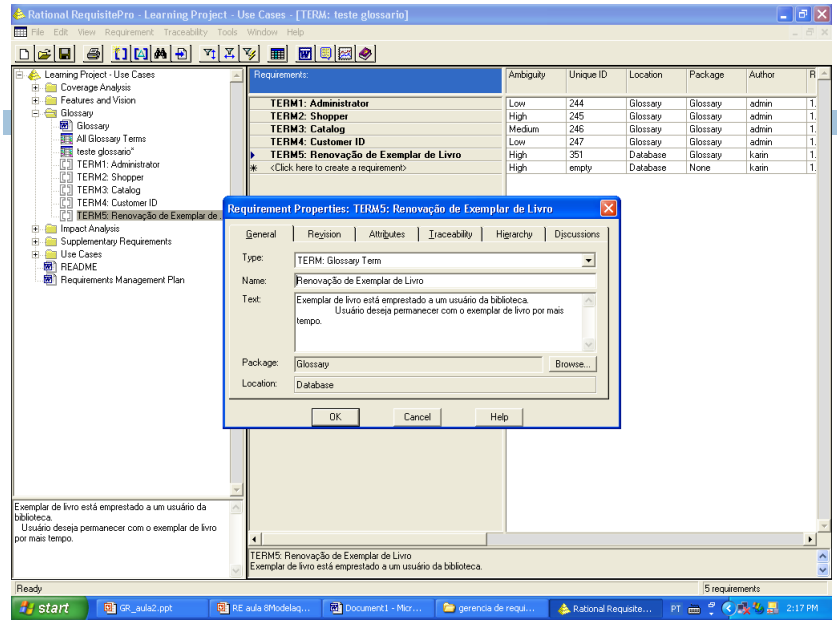


© KKB/PUC-Rio

Glosário

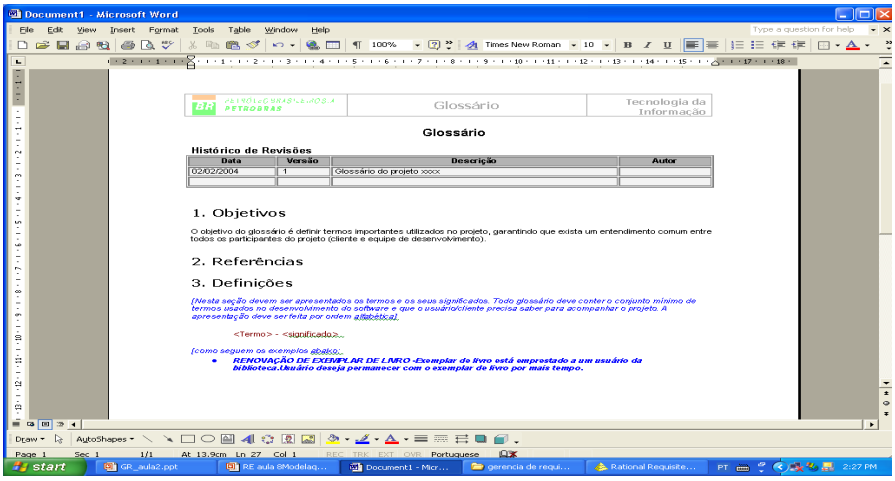
- Requisite Pro
  - Requisito do tipo TERM:Glossary term
  - Pertence ao Pacote Glossary
  - Contém significado apenas

© KKB/PUC-Rio



Glossário – Template Petrobrás

283


Document1 - Microsoft Word
Glossário
Tecnologia da Informação
Glossário
Histórico de Revisões
Data
Versão
Descrição
Autor
02/02/2004
1
Glossário do projeto xxxx
1. Objetivos
O objetivo do glossário é definir termos importantes utilizados no projeto, garantindo que exista um entendimento comum entre todos os participantes do projeto (cliente e equipe de desenvolvimento).
2. Referências
3. Definições
[Nesta seção devem ser apresentados os termos e os seus significados. Todo glossário deve conter o conjunto mínimo de termos usados no desenvolvimento do software e que o usuário/cliente precisa saber para acompanhar o projeto. A apresentação deve ser feita por ordem alfabética].
<Termo> - <significado>.
[como seguem os exemplos abaixo:]
RENOVAÇÃO DE EXEMPLAR DE LIVRO -Exemplar de livro está emprestado a um usuário da biblioteca.Usuário deseja permanecer com o exemplar de livro por mais tempo.

© KKB/PUC-Rio

Léxicos e Glossários

284

- \+
  - Reduz ambiguidade
  - Evita "mal entendidos"'
  - Aumenta a precisão da especificação
  - Melhora a comunicação entre os interessados

© KKB/PUC-Rio

Léxicos e Glossários

285

- -
  - Consome tempo
  - necessita validação

© KKB/PUC-Rio

Qualidade de modelos

7 Pecados [Myers]

- Informações irrelevantes
- Incompleteza
- Excesso de Especificação (≠ decisões de desenho)
- Inconsistência
- Ambiguidade
- Referência futura
- Requisitos que não podem ser testados

© KKB/PUC-Rio

Armazenamento

287

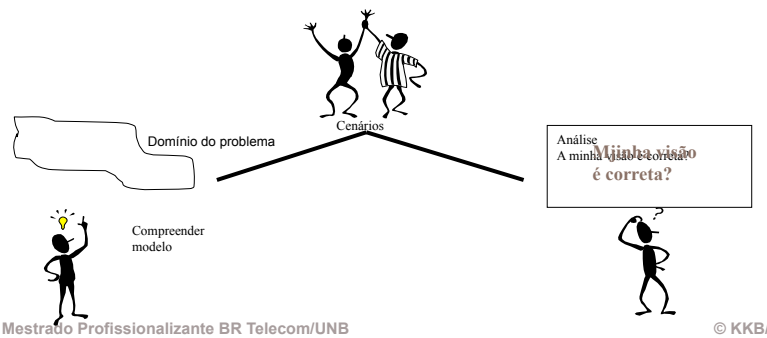
- Para que a modelagem seja efetiva é preciso que seja armazenada a contento, e principalmente que possa ser recuperada quando for necessário.
- Para que o armazenamento seja eficaz é indispensável que exista:
  - Classificação,
  - Indexacão e
  - Apresentação.
- Esquemas de facetas (Reuso) : Facetas funcionais (função, objeto, meio), Facetas Não-Funcionais (tipo do sistema, área funcional, contexto).

© KKB/PUC-Rio

Comunicação

288

- Necessidade de garantir boa comunicação entre clientes/usuários e desenvolvedores



© KKB/PUC-Rio

Cenários

289

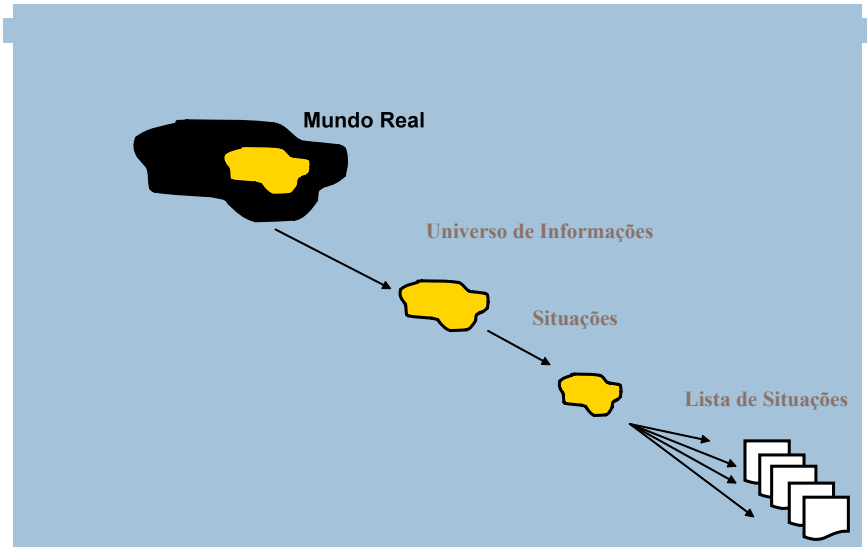
- Fáceis de entender (escritos na linguagem do problema)
- Ajudam a unificar critérios
- Estimulam o pensamento
- Ajudam no treinamento
- Ajudam a manter rastreabilidade
- Ajudam na identificação de requisitos não-funcionais

Cenários são situações

© KKB/PUC-Rio

Cenários

290



© KKB/PUC-Rio

Escrever Cenários

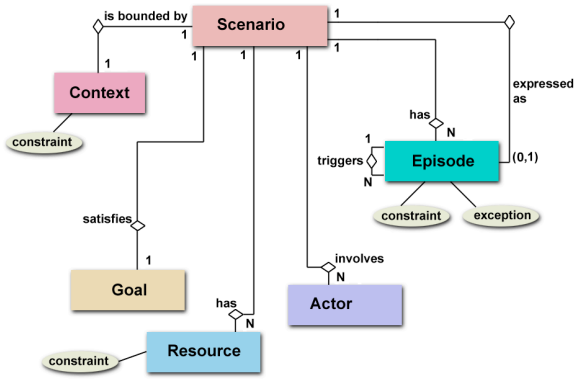
291

- Descrever situações do macrosistema
- Descrever situações e seus relacionamentos com o sistemas a ser construído
- Podem ser utilizados para descrever a interação entre componentes do sistema
- Utilizam linguagem natural **semi-estruturada**

© KKB/PUC-Rio

Cenários

292



© KKB/PUC-Rio

Porque Semi Estruturada?

293

- Evita confusão
- Garante um estilo de descrição homogênio
- Serve como lembrete dos vários aspectos que devem ser comtemplados no cenário.
- Facilita validação com fregueses.

© KKB/PUC-Rio

Cenários

294

Título: Loja confere ficha cadastral

Objetivo: Verificar se as informações da ficha cadastral estão corretas

Contexto: Cliente entrega ficha cadastral e apresenta carteira de identidade e CPF para loja

Atores: loja.

Recursos: carteira de identidade, CPF, ficha cadastral Restrição: deve ter precisão de propriedade e precisão de valor.

Episódios

Loja preenche os campos da ficha cadastral não informados pelo cliente com a sigla 'NE' (não existente)

Restrição: ficha cadastral deve ter precisão de propriedade, sendo a estratégia de satisfação validação

Loja verifica o número da identidade presente na ficha cadastral com a carteira de identidade do cliente

Restrição: ficha cadastral deve ter precisão de valor, sendo a estratégia de satisfação validação

Loja verifica o número de CPF presente na ficha cadastral com o CPF do cliente

Restrição: ficha cadastral deve ter precisão de valor, sendo a estratégia de satisfação validação

Loja verifica o endereço e o telefone de contato presentes na ficha cadastral, ligando para o telefone de contato

Restrição: ficha cadastral deve ter precisão de valor, sendo a estratégia de satisfação validação

© KKB/PUC-Rio

Cenários

295

Título: Adicionar exemplar de livro ao acervo da biblioteca

Objetivo: Exemplar de livro pertencer ao acervo da biblioteca

Contexto: Quantidade de exemplares de livro pertencentes ao acervo da biblioteca insuficiente.

Há espaço físico para colocar novo exemplar de livro no acervo da biblioteca.

Exemplar de livro pode ser comprado ou recebido através de doação.

Funcionário da biblioteca está presente na biblioteca.

Sistema de gerenciamento da biblioteca está funcionando.

Atores: funcionário da biblioteca

Recursos: exemplar de livro, livro, acervo da biblioteca, sistema de gerenciamento da biblioteca

Episódios:

1. Funcionário da biblioteca recebe exemplar de livro a ser adicionado ao acervo da biblioteca.
2. Se os dados do livro ainda não foram cadastrados no sistema de gerenciamento da biblioteca, funcionário da biblioteca deve CADASTRAR LIVRO NO ACERVO DA BIBLIOTECA.
3. Funcionário da biblioteca cadastra exemplar de livro no sistema de gerenciamento da biblioteca.
4. Funcionário da biblioteca aloca espaço físico no acervo da biblioteca para colocar exemplar de livro de acordo com informações retornadas pelo sistema de gerenciamento da biblioteca, levando em conta o arranjo do espaço físico do acervo da biblioteca.
5. Funcionário da biblioteca coloca exemplar de livro no acervo da biblioteca.

Paulo Roberto Bastos
Juliana Carpes Imperial
Elton Silva

© KKB/PUC-Rio

Identificação de Cenários

296

- Listar situações
  - 1. existe um objetivo? - é geral o suficiente? existem saídas diferentes, ou se trata de apenas um caso?
  - 2. quem está envolvido? - existem outros artefatos ou estruturas importantes?
  - 3. Existem informações ou elementos físicos importantes para esta situação?
  - 4. Organize as situações identificadas em uma lista.

© KKB/PUC-Rio

Preencher os Cenários

297

- Não invente !!!
- Utilize o vocabulário da aplicação
- Utilizando a gramática de cenários preencha os cenários candidatos (*pair programming* com cliente é sempre melhor)
- Utilize termos do léxico

© KKB/PUC-Rio

Estilo

298

- Frases curtas
- Evitar mais de um verbo por frase
- O objetivo deve ser concreto e preciso
- Pelo menos um dos componentes do contexto deve estar preenchido
- Os recursos devem ser aqueles diretamente involvidos nos episódios. Evite triavialidades.

© KKB/PUC-Rio

Organização

299

- Léxico --> hypertexto
- Cenários --> Relações (complemento, pre-condição, equivalente, exceção, sub-conjunto, possível precedência, inclusão).
- Sentenças (itemização númerica, capítulos)

© KKB/PUC-Rio

Cénarios - Processo

300

- **Derivar**
  - **identificar atores**
  - **identificar cenários**
  - **criar candidatos**
- **Descrever**
  - **utilizar representação**
  - **seguir "dicas"**
  - **juntar cenários**
- **Organizar**
  - **reorganizar**
  - **definir**
  - **integrar**
- **Verificar**
- **Validar**

© KKB/PUC-Rio

Cénarios

301

- Frases curtas
- Maximize o uso de símbolos do Léxico
- Use apenas um verbo por frase.
- Atores e recursos devem ser símbolos do léxico.
- O objetivo deve ser concreto e preciso.
- O contexto deve ter pelo menos um ítem (local, tempo, pre-condição).
- Recursos devem listar todos os recursos utilizados nos episódios, exceto aqueles utilizados em sub-cenários.
- Atores devem listar todos aqueles envolvidos em episódios, exceto aqueles utilizados em sub-cenários.
- O verbo de cada episódio deve ser pontual.
- Os episódios devem acontecer nos limites/restrições impostas pelo contexto.
- Evite utilizar verbos como: “poderia”, “controla”, “deve” ...

© KKB/PUC-Rio

Sentenças de Requisitos

302

- Derivar dos Cenários
- Classificar quanto ao Tipo
- Enumerar (organização)

© KKB/PUC-Rio

Casos de Uso

303

- Oferecem uma maneira intuitiva e sistemática para capturar os requisitos FUNCIONAIS
- Foco no conceito de "valor adicionado" (*added value*) ao cliente
- Podem ser utilizados para guiar o processo de desenvolvimento

[Unified Software Development Process – Jacobson, Booch, Rumbaugh]

© KKB/PUC-Rio

"Valor adicionado"

304

- Perguntar aos clientes/usuários o que eles gostariam que o sistema fizesse não funciona:
  - Lista de funcionalidades que pode parecer útil a princípio, mas na verdade...
    - Ex: modificar endereço da cobrança
- Estratégia de Caso de Uso:
  - Refrazear para "O que você quer que o sistema faça *PARA CADA USUÁRIO*?

© KKB/PUC-Rio

Casos de Uso

- **Um caso de uso realiza um aspecto maior da funcionalidade do produto:**
  - **deve gerar um ou mais benefícios para o cliente ou para os usuários**
  - **Podem representar:**
    - **roteiros de interação com usuário**
    - **roteiros do manual de usuário**
    - **casos de teste**

Filho, W.P.P em "Engenharia de Software: Fundamentos, Métodos e Padrões"

© KKB/PUC-Rio

O que deve estar no caso de uso?

- Casos de uso descrevem um sistema
  - Se estiver utilizando casos de uso para descrever os processos e regras de negócio de uma empresa – a própria empresa é o sistema em questão.
  - Se estiver utilizando casos de uso para descrever um software – o programa é o sistema em questão.

© KKB/PUC-Rio

O que deve estar no caso de uso?

307

- Escopo – qual o sistema em questão?
- Ator principal – quem tem o objetivo?
- Nível – quão alto ou baixo nível é o objetivo?

© KKB/PUC-Rio

Representação

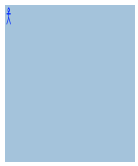
308

- Casos de Uso são fundamentalmente textuais (também podem ser representados através de flow charts, diagramas de sequência, redes de Petri e diagramas de estado)
- Diagramas de Caso de uso – representação gráfica (mostra os atores e os casos de uso em que estão envolvidos)

© KKB/PUC-Rio

Diagrama de Caso de Uso Sintaxe

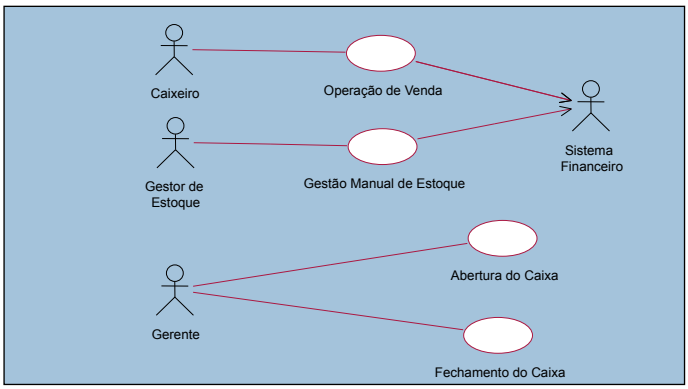
- Ator (Actor)



- Caso de Uso (Use Case)

© KKB/PUC-Rio

Casos de Uso



© KKB/PUC-Rio

Casos de Uso [Cockburn]

311

- Comprar ações na Web

Escopo: conselheiro/pacote financeiro(PAF)

Nível: objetivo do usuário

Interessados e interesses:

comprador- quer comprar ações e tê-las adicionadas ao portfólio PAF automaticamente

Financeira – quer informação de compra

Pré condição: usuário tem o PAF aberto

Garantia mínima: informação de log suficiente de modo que o PAF possa detectar erros e solicitar mais detalhes

Garantia de sucesso: Site remoto tem conhecimento da compra, dos logs e o portfolio do usuário é atualizado

Cenário de Sucesso – Principal:

1. Comprador seleciona ações na internet
2. PAF pega nome do web site a ser utilizado (Schwab, E trade)
3. PAF abre conexão para o site, retendo o controle
4. Comprador navega e compra ação do site
5. PAF intercepta respostas do site da web e atualiza portfolio
6. PAF mostra o novo portfolio

Extensões:

2a. Comprador seleciona um site que o PAF não trabalha

2a1. Sistema recebe novas sugestões do comprador, com opção de cancelar o caso de uso

Mestrado Profissionalizante BR Telecom/UNB © KKB/PUC-Rio

Caso de Uso [Constantine e Lockwood]

Cliente	Sistema
Entra número da ordem de serviço (OS)	
	Detecta que o número da OS casa com o número do vencedor do mês
	Registra o número da OS como vencedora do mês
	Manda mensagem de e-mail para o representante de vendas
	Congratula o cliente e fornece instruções para a coleta do prêmio
Sai do sistema	

Casos de Uso [W.P.P. Filho]

313

<< Operação de Venda>>

- **pre-condições:** Toda mercadoria a ser vendida (item de venda) deve estar previamente cadastrada. O Merci deve estar em Modo de Vendas.
- **fluxo principal**
  - O **Caixeiro** faz a abertura da venda.
  - O **Merci** gera o código da operação de venda.
  - Para cada item de venda aciona o subfluxo **Registro**.
  - O **Caixeiro** registra a forma de pagamento.
  - O **Caixeiro** encerra a venda.
  - Para cada item aciona o subfluxo **Impressão de Linha do Ticket**.
  - O **Merci** notifica o **Sistema Financeiro** informando: Data, Número da Operação de Venda, "Receita", Valor Total", Nome do Cliente (caso tenha sido emitida a nota fiscal).

© KKB/PUC-Rio

Casos de Uso (RUP)

314

1. Nome
   - Descrição
   - Atores
   - Disparadores
2. Fluxo de eventos
   - Fluxo básico
   - Fluxo alternativo
     - Condição 1
     - Condição 2
3. Requisitos Especiais
   - Plataforma
4. Pré condições
5. Pós Condições
6. Pontos de Extensão

© KKB/PUC-Rio

Template Petrobras

315

- <Nome do Caso de Uso>
- 1. Breve Descrição
- 2. Atores
- 3. Fluxos de Eventos
  - 3.1. Fluxo Básico
  - 3.2. Fluxos Alternativos
- 4. Requisitos Especiais
- 5. Precondições
- 6. Pós-Condições
- 7. Pontos de Extensão
- 8. Outros Diagramas

© KKB/PUC-Rio

NOME: **Organizar a reunião**

DESCRIÇÃO: Processo de planejamento e organização para garantir a eficiência da reunião

ATORES :iniciador da reunião, secretária, participantes.

FLUXO BÁSICO:

O iniciador da reunião fornece a sua secretária a lista de participantes e intervalo de datas para que ela faça a chamada para reunião.

Secretária faz chamada para reunião aos participantes. Secretária centraliza as notificações de presença dos participantes.

Secretária requisita sala de reunião.

Secretária requisita equipamento necessário.

SE a data da reunião for com antecedência de uma semana **ENTÃO** lembrar participantes.

FLUXO ALTERNATIVO:

Iniciador ou participantes não poderão estar presentes na reunião (cancelar reunião)

A data da reunião deve ser modificada (modificar data de reunião).

REQUISITOS ESPECIAIS:

[A secretária garante que a sala de reunião esteja disponível na data da reunião]

[A secretária garante que o equipamento solicitado esteja disponível na data da reunião]

PRÉ CONDIÇÕES

A reunião foi marcada previamente.

PÓS CONDIÇÕES

Não foram registradas pós condições para este caso de uso

PONTOS DE EXTENSÃO - Não foram registrados outros diagramas para este caso de uso

OUTROS DIAGRAMAS - Não foram registrados outros diagramas para este caso de uso

316 **Mestrado Profissionalizante BR Telecom/UNB**
© KKB/PUC-Rio

Exercício - Caso de Uso

317

Pedir promoção no McDonald's

- Fluxo básico:

1. O caso de uso inicia quando chega a vez do cliente na fila do caixa.
2. (seleciona promoção).
3. (bebida).
4. Funcionário oferece a opção de aumento de B&B.
5. (sobremesa).
6. Cliente decide se o pedido é para viagem ou para agora.

Dicionário:
•Promoção
Combinação de sanduíche, batata e refrigerante de 300ml
•Pedido
•Opção de aumento de B&B
•Pedido para viagem – para viagem
Pedido servido em sacos de papel. Inclui guardanapos, canudos e porções de ketchup e mostarda
•Pedido para agora – para agora
•Sobremesa

© KKB/PUC-Rio

Caso de Uso

318

- Pedir promoção no McDonald's

Fluxo básico:

1. O caso de uso inicia quando chega a vez do cliente na fila do caixa.
2. Cliente seleciona número da promoção.
3. Cliente seleciona tipo de bebida.
4. Funcionário oferece a opção de aumento de bebida/batata.
5. Funcionário oferece sobremesa.
6. Cliente decide se o pedido é para viagem ou para agora.

© KKB/PUC-Rio

Caso de Uso

Cenários Secundários

- Requisitos especiais (sem pickles, sem gelo, molho extra)
- Demora no sanduíche
- Ítem indisponível
- Promoção sazonal (sanduíche especial, sobremesas)

© KKB/PUC-Rio

Exercício

- Escrever um caso de uso para a seguinte situação:
- Para fazer um pedido um *usuário* precisa se identificar (indicar nome e endereço de entrega), indicar o código dos produtos que deseja comprar e informações de pagamento (cartão de crédito – nome completo, número e validade). O sistema valida a informação de pagamento do *cliente*, e, se confirmada, fornece número do *pedido* para o cliente.
- Quando uma *ordem* é submetida, recebe o status de pendente. O *pedido* só deve ser confirmado quando a informação de pagamento for confirmada.
- Seria interessante que o sistema mantivesse o *usuário* sempre informado do total de seu *pedido*

© KKB/PUC-Rio

Requisitos e Casos de Uso

321

- Casos de Uso são requisitos – não é necessário transformar para outro formato. Se escritos com cuidado, os casos de uso detalham o que o sistema deve fazer
- Casos de uso não são TODOS os requisitos – eles NÃO detalham as interfaces externas, formatos de dados, regras de negócio... Eles constituem apenas uma FRAÇÃO dos requisitos de um projeto

[Writing Effective Use Cases – Alistair Cockburn]

© KKB/PUC-Rio

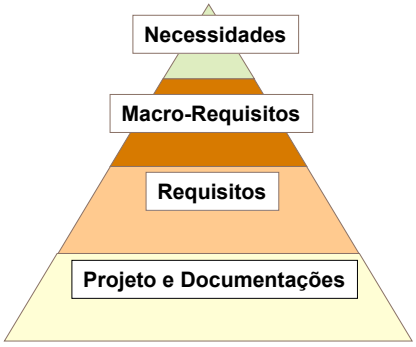
Dimensões

322

- Escopo dos casos de uso
  - Organização, Sistema, Componente
- Granularidade dos casos de uso – nível de detalhamento dos objetivos: alto nível, geral, objetivo de usuário, subfunção
- Processo - Iterações sucessivas – detalhar aos poucos para evitar gasto desnecessário de recursos

© KKB/PUC-Rio

Escopo dos Casos de Uso



Domínio do problema

Domínio da solução

Organização

Sistema

Componente

© KKB/PUC-Rio

Objetivos

- Procurar casos de uso que agregam valor
  - Evitar Casos de Abuso (misuse case)
- Evitar detalhamento excessivo
  - Casos sem Uso (useless case)

© KKB/PUC-Rio

"Não há sentido em ser preciso se nem sabemos do que estamos falando"

- Evitar detalhes no início
  - Dar uma chance para os usuários fazerem correções e priorizarem suas necessidades
  - Permitir que o trabalho possa ser dividido e feito em paralelo

© KKB/PUC-Rio

Níveis de precisão/detalhamento

- Listar atores e objetivos (diagrama de casos de uso)
- Descrição breve do caso de uso - Para os casos de uso selecionados como prioritários, descrever brevemente o fluxo básico (sucesso)
- Falhas – completar o fluxo básico e brainstorm condições de falhas
- Resposta a falhas – detalhar a resposta do sistema a falhas

© KKB/PUC-Rio

Elicitação de Casos de Uso

327

- Identificação de atores;
  - quem está interessado nos requisitos;
  - quem se beneficiará do produto;
  - quem fornecerá informação ao produto;
  - quem usará informação do produto;
  - quem removerá informação do produto.
- Identificação dos casos de uso:
  - Quais os objetivos de cada caso de uso?
  - Qual o escopo do caso de uso?
  - Que informação cada caso de uso cria, armazena, consulta, altera ou remove.

© KKB/PUC-Rio

Elicitação de Casos de Uso

328

- Determinação do fluxo de um caso de uso
  - Quando e como o caso de uso principia.
  - Como o caso de uso interage com os atores.
  - Sequência padrão dos passos do caso de uso.
  - Sequências de exceção e alternativas dos passos do caso de uso.
  - Existem extensões?
- Técnicas de elicitação
  - Questionários
  - Entrevistas
  - Observação
  - Reuniões .....

© KKB/PUC-Rio

Priorização de Requisitos

© KKB/PUC-Rio

O que é priorizar? [Wiegers]

- Trade off entre:
  - escopo
  - tempo
  - recursos
- Garantir que o **essencial** é realizado

© KKB/PUC-Rio

Por que priorizar?

331

- Expectativas altas
- Tempo curto
- Recursos limitados

- Ditado:

"*agente faz o que pode e não o que quer*"

© KKB/PUC-Rio

Porque priorizar?

332

- Controlar o escopo do projeto: Síndrome do "já que"
- Caper Jones reporta que os requisitos que "rastejam para debaixo"do escopo representam
  - risco de 80% a projetos de gerência de informação
  - risco de 70% a projetos militares

© KKB/PUC-Rio

Porque priorizar?

333

- Requisitos rastejantes
  - nova funcionalidade
  - modificações
    - requisitos
    - aumento do escopo

Diga NÃO !!!

© KKB/PUC-Rio

Técnicas de priorização

334

- Formais
  - QFD
- Informais
  - R$ 100
  - Categorizar

© KKB/PUC-Rio

QFD (quality function deployment)

335

- Participantes:
  - gerente
  - figuras chave
  - desenvolvedores

[Cohen95/Wiegers]

© KKB/PUC-Rio

QFD

336

- Etapas do processo:

1. Listar em uma planilha os requisitos, funcionalidade ou casos de uso que se deseja priorizar
2. Estimar o benefício relativo de cada um utilizando uma escala de 1 - 9, onde 1 representa um benefício negligenciável e 9 representa grande valor.

© KKB/PUC-Rio

QFD

337

3. Estimar a penalidade o freguês ou negócio sofreria se o ítem não fosse incluído. Utilize a escala de 1-9, onde 1 representa uma penalidade ínfima e 9 grande perda
4. Crie uma coluna - valor total - que é a soma do benefício com a penalidade
5. Utilize a planilha para calcular a percentagem do valor total que cada ítem representa (valor%)

© KKB/PUC-Rio

QFD

338

6. Estime o custo de implementação de cada ítem também utilizando uma escala de 1(baixo) a 9 (alto) - (custo)
7. Utilize a planilha para calcular a percentagem do custo total que cada ítem representa - (custo%)
8. Estime o risco que cada ítem representa utilizando a escala de 1-9 - (risco)
9. Utilize a planilha para calcular a percentagem do risco total que cada ítem representa - (risco %)

© KKB/PUC-Rio

QFD

339

- Uma vez que se tenha todas as estimativas:

$$\text{Prioridade} = \frac{\text{valor\%}}{(\text{custo\%}*\text{custo})+(\text{risco\%}*\text{risco})}$$

- Organize os ítens em forma decrescente, por prioridade

© KKB/PUC-Rio

QFD

340

Crítica:

- Método semiquantitativo
- Não rigoroso
- Depende da abilidade pessoal de realizar estimativas de benefício, custo e risco.

© KKB/PUC-Rio

Técnicas informais [Leffingwell]

341

R$100

- durante uma reunião
- cada participante recebe R$100 para gasto na compra de requisitos
- escrever em um papel quanto do dinheiro gastaria em cada requisito
- tabular resultados
- ranking dos requisitos

© KKB/PUC-Rio

Técnicas informais

342

Categorização

- off line
- cada interessado recebe a lista dos requisitos

© KKB/PUC-Rio

Técnicas informais

343

Categorização

- classifica segundo os critérios
  - crítico - indispensável
  - importante - representa perda de funcionalidade ou perda de fatia do mercado
  - útil - facilita a vida, faz o sistema mais atraente
- atribui valores 1,2,3 (onde 1 é crítico)
- Fazer um ranking dos resultados

© KKB/PUC-Rio

Outras escalas de priorização de requisitos: [Wiegers]

344

IEEE 1998

essencial

software não é aceitável a não ser que estes requisitos sejam implementados

condicional

melhoraria produto, mas não o tornaria inaceitável se ausente

opcional

classe de funcionalidade que pode ou não valer a pena

Outras escalas de priorização de requisitos: [Wiegers]

345

Kovitz 1999

3 - dever ser implementado de modo perfeito

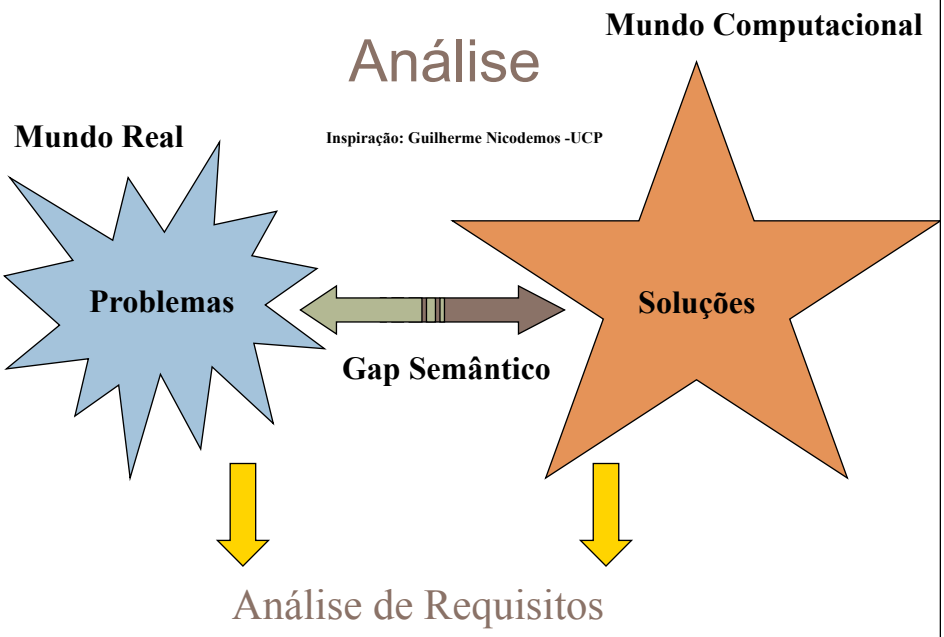
2 - precisa funcionar, mas não de modo espetacular

1 - pode conter bugs

346

Análise

© KKB/PUC-Rio



© KKB/PUC-Rio

Análise

- Identificação das partes
  - como está organizado?
  - como está armazenado?
  - quem e o quê compõem as partes?
  - rastreamento

© KKB/PUC-Rio

Análise

349

- Verificação
  - formal
  - reusa domínios
  - inspeções

© KKB/PUC-Rio

Análise

350

- Validação
  - junto aos usuários/fregueses
  - informal
  - prototipagem (mock up, storyboard)
  - pontos de vista

© KKB/PUC-Rio

Verificação X Validação

351

Verificação	Validação
estamos construindo o produto de maneira certa?	estamos construindo o produto certo?
(em relação a outros produtos)	(em relação a intenção dos fregueses)
↓	↓
entre modelos	entre o Udl e um modelo

Identificação de partes

352

- Depende da organização e armazenamento
- é ligada a identificação das fontes de informação
- 90% dos problemas em 10% do sistema
- museu britânico X heurísticas
- enfoque estatístico

Validação

353

- Estamos construíndo o produto certo?
- Acontece com a comparação de expectativas dos atores do Udl.

© KKB/PUC-Rio

Validação

354

- Teste
- Execução de cenários (leitura em reunião)
- Leituras por atores interessados.
- Protótipos

© KKB/PUC-Rio

Estratégias de validação

355

- Usando comprovação informal
- storyboards
- protótipos
- formalismo
- reusando domínios
- pontos de vista

© KKB/PUC-Rio

Storyboard [Leffingwell & Widrig]

356

- Elicitar reações do tipo "sim, mas..."
- passivo, ativo ou interativo
- identifica **atores**, explica **o que** acontece a eles e descreve **como** acontece

© KKB/PUC-Rio

Storyboard [Leffingwell & Widrig]

357

- mais eficazes se o projeto tiver conteúdo inovador ou desconhecido
- tipo rascunho, fáceis de modificar
  - princípio da "negação construída)

© KKB/PUC-Rio

Storyboard

358

- Vantagens:
  - barato
  - amigável, informal e interativo
  - fornece uma crítica das interfaces do sistema cedo no desenvolvimento
  - fácil de criar e modificar

© KKB/PUC-Rio

Tipos de Storyboard

359

- Passivo
  - telas estáticas
  - regras de negócio
  - relatórios

© KKB/PUC-Rio

Tipos de Storyboard

360

- Ativo
  - apresentação (tipo PowerPoint)
  - animação
  - simulação

© KKB/PUC-Rio

Tipos de Storyboard

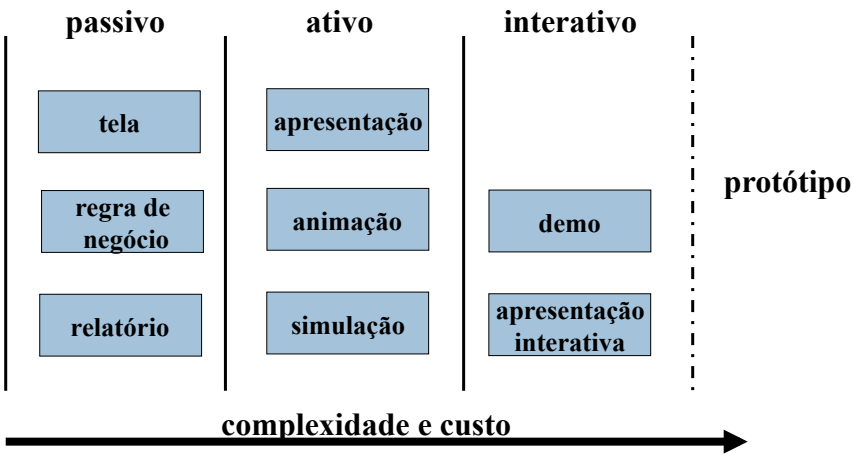
361

- Interativo
  - demo (navegação livre)
  - apresentação interativa

© KKB/PUC-Rio

Storyboard

362



Protótipos

363

- protótipos são implementações parciais para auxiliar fregueses, usuários e desenvolvedores a melhor entender os requisitos do sistema

© KKB/PUC-Rio

Tipos de protótipo [Davis]

364

- Jogar fora
  - tem que funcionar
  - utiliza qualquer artifício para implementar o resultado desejado (gato, bacalhau..)
  - uma vez levantados os requisitos o protótipo é jogado fora

© KKB/PUC-Rio

Tipos de protótipo [Davis]

365

- Evolutivo
  - implementado utilizando a mesma arquitetura que será usada no sistema
  - o sistema poderá ser construído como uma evolução deste protótipo

© KKB/PUC-Rio

Protótipos

366

Vertical X Horizontal

- Horizontal
  - implementa grande parte da funcionalidade
- Vertical
  - implementa poucas funções
  - maior qualidade

© KKB/PUC-Rio

Verificação

367

- Estamos construído o produto de maneira correta?
- Acontece com a utilização de duas entidades modeladas
- Acontece com a utilização de conhecimento empacotado.

© KKB/PUC-Rio

Verificação

368

- Pouca ênfase na verificação
- Escolher a sub-divisão em que se vai trabalhar
- Antecipar os recursos e atores do Udl necessários para levar a cabo a tarefa.

© KKB/PUC-Rio

Verificação

369

- Uso de modelos
  - representações/linguagens
- Uso de formalismos
- Reuso de formalismos
- Técnicas informais

© KKB/PUC-Rio

Uso de formalismos

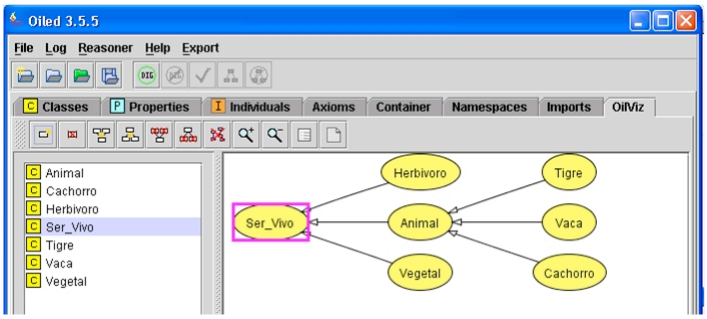
370

- Provas formais de um modelo
  - implementa/reifica o outro (provador de teoremas)
- Detecção de discrepâncias entre o modelo e meta modelo
  - verificadores de modelos
- Detecção de discrepâncias sintáticas entre modelo e meta modelo
  - analisadores sintáticos, editores

© KKB/PUC-Rio

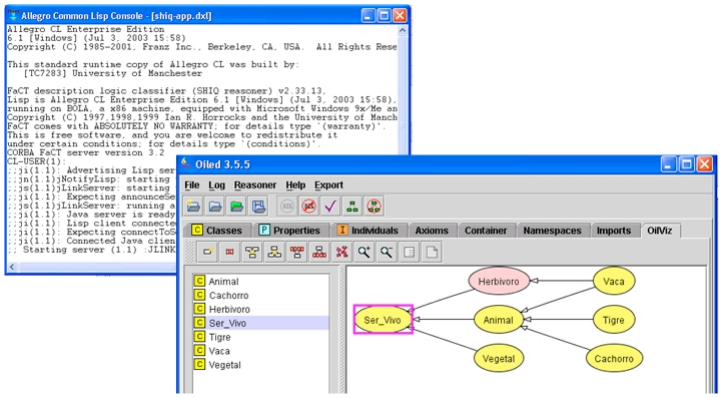
Inferência

371



Inferência

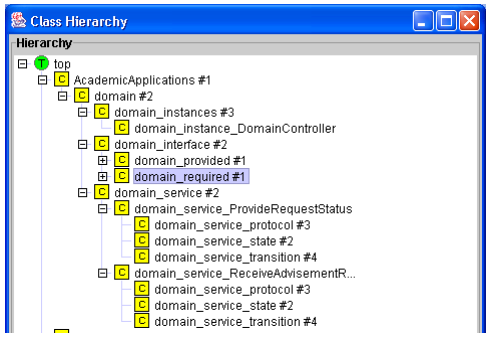
372



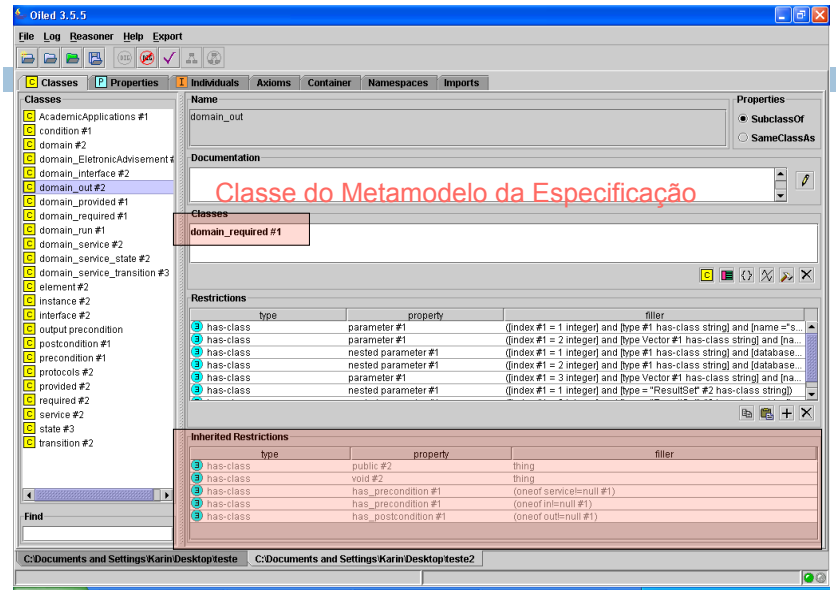
© KKB/PUC-Rio

Especificação Formal

```
- <Domain_name="AcademicApplications">
  - <Interface>
      + <Provided>
      + <Required>
    </Interface>
  - <Service>
      + <Service_name="Receive
            Advisement Requests">
      + <Service_name="Provide Request
            Status">
    </Service>
  - <Instance>
      + <DomainController>
      + <MAS_name="Electronic
            Advisement">
    </Instance>
    </Domain_name>
```



Informal - XML → Formal - Ontologia

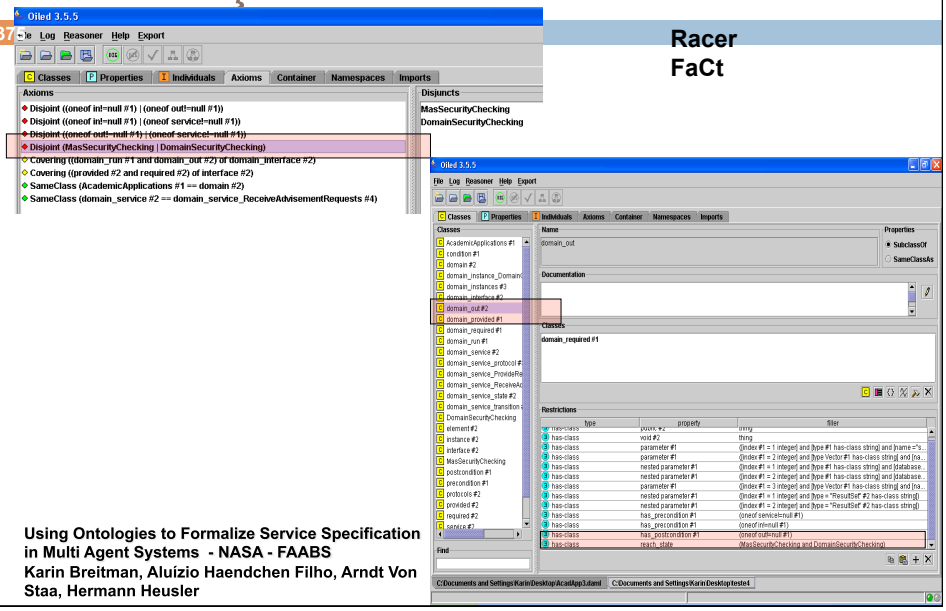


© KKB/PUC-Rio

Verificação automática

37

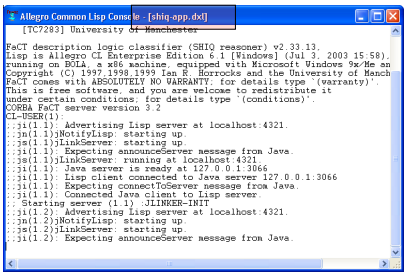
Racer
FaCt

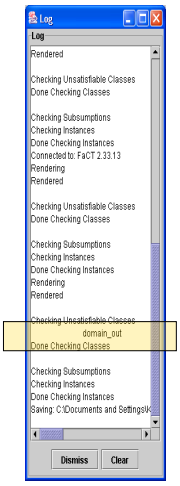

Oiled 3.5.5
File Log Reasoner Help Export
Classes Properties Individuals Axioms Container Namespaces Imports
Axioms
Disjoint ((oneof in!=null #1) | (oneof out!=null #1))
Disjoint ((oneof in!=null #1) | (oneof service!=null #1))
Disjoint ((oneof out!=null #1) | (oneof service!=null #1))
Disjoint (MasSecurityChecking | DomainSecurityChecking)
Covering ((domain_run #1 and domain_out #2) of domain_interface #2)
Covering ((provided #2 and required #2) of interface #2)
SameClass (AcademicApplications #1 == domain #2)
SameClass (domain_service #2 == domain_service_ReceiveAdvisementRequests #4)
Disjuncts
MasSecurityChecking
DomainSecurityChecking

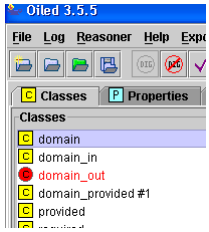
Using Ontologies to Formalize Service Specification in Multi Agent Systems - NASA - FAABS
Karin Breitman, Aluízio Haendchen Filho, Arndt Von Staa, Hermann Heusler

Verificação Automática

376


Allegro Common Lisp Console - [shiq-app.dxl]
FaCT description logic classifier (SHIQ reasoner) v2.33.13.
Lisp is Allegro CL Enterprise Edition 6.1 [Windows] (Jul 3, 2003 15:58).
FaCT comes with ABSOLUTELY NO WARRANTY: for details type '(warranty)'.
This is free software, and you are welcome to redistribute it
under certain conditions; for details type '(conditions)'.
CORBA FaCT server version 3.2
CL-USER(1)





Using Ontologies to Formalize Service Specification in Multi Agent Systems - NASA - FAABS
Karin Breitman, Aluízio Haendchen Filho, Arndt Von

Model Checking

377 [Jackson, D.]

- O que é?
  - técnica para analizar máquinas de estado
- Quem usa?
  - originalmente para verificação de hardware
  - primeiros model checkers na década de 80
  - estão sendo adotados pela indústria de semicondutores
  - utilização em software

© KKB/PUC-Rio

Model Checking

378 [Jackson, D.]

- Porque é melhor do que outros métodos convencionais?
  - totalmente automático
  - rápido
  - análise exaustiva - nenhum erro fica de fora
  - pode manipular máquinas enormes (1bilhão de estados)
  - apresenta contraexemplos quando detecta falhas

© KKB/PUC-Rio

Reuso de Domínios

379

- Com base em “teoria codificada”
- Verificação de discrepâncias entre modelo proposto (requisitos) e “teoria”
- Pressupõe detecção de conflito e ausência
- Defende a qualidade da “teoria”

© KKB/PUC-Rio

Reuso de Domínios

380

- Com base em “teoria codificada”
- Verificação de discrepâncias entre modelo proposto (requisitos) e “teoria”
- Pressupõe detecção de conflito e ausência
- Defende a qualidade da “teoria”

© KKB/PUC-Rio

Técnicas informais

381

- Leitura ad hoc
- Walk throughs
- Inspeções

© KKB/PUC-Rio

Walk through
(revisão de apresentação)

382

- Preparação ad hoc
- Reunião (autor(es), avaliador(es), secretário(s))

© KKB/PUC-Rio

Walk through
(revisão de apresentação)

383

- Leitura
  - autor lê
  - avaliadores escutam
  - avaliadores apontam problemas (perguntas)
  - secretário(s) anotam problemas
- Lista de problemas

© KKB/PUC-Rio

Checklist

384

- Pontos a serem avaliados/observados durante o processo de inspeção
- Depende do material a ser inspecionado (DFD, cenários, JSD...)
- Depende do enfoque da inspeção
- Taxonomia dos defeitos - o que procurar

© KKB/PUC-Rio

Exemplo DFD

385

- Checklist DFD
  - a documentação apresentada deve ter:
    - data, páginas numeradas, lista de tópicos, controle de mudanças e versões com indicação dos responsáveis.
  - Processo representado por círculo numerado
  - Identificador deve começar com verbo
  - O diagrama de contexto deve ter relacionamentos com etiquetas e depósitos de dados externos ao sistema
  - O número máximo de processos deve ser 7 +- 2

© KKB/PUC-Rio

Exemplo OO

386

- Checklist OO:
  - Todas as classes são representadas por retângulos com 1,2 ou 3 compartimentos?
  - As classes possuem nomes diferentes?
  - Existem classes sem relacionamentos definidos?
  - Os atributos e os métodos de cada classe são adequados aquela classe?
  - Todo comportamento especificado é possível de ser contemplado através das associações do modelo?

© KKB/PUC-Rio

Exemplo Diagrama de Casos de uso (Petrobras)

IDENTIFICADOR	GRUPO	ITEM DE VERIFICAÇÃO
1.		Todo ator deve estar associado a pelo menos um caso de uso. (se existe um ator que não interage ele não é ator)
2.		Todo caso de uso concreto deve ser iniciado por um ator
3.		Só são permitidos 3 tipos de associação entre casos de uso: <<include>>, <<extend>> e <<generalization>> (verificar superconjunto do RUP)
4.		Casos de uso que fazem parte de pacotes (agrupamentos) diferentes devem fazer parte de diagramas diferentes.

© KKB/PUC-Rio

Exemplo Esp. Casos de Uso

1. Deve ser seguido o template do documento "Especificação do Caso de Uso
2. <Nome do Caso de Uso> "
3. O nome do caso de uso deve começar com um verbo
4. Deve ter apenas um fluxo básico
5. seus passos devem ser numerados sequencialmente (1,2,3,...)
6. no fluxo básico não se faz referências aos fluxos alternativos
7. Cada fluxo alternativo deve conter as seguintes informações:

© KKB/PUC-Rio

Taxonomia dos enfoques de inspeção

389

- Baseado em cenários
- Leitura construtiva
- Baseado em checklist

© KKB/PUC-Rio

Documento de Requisitos: inspeções

390

- Inspeções:
  - ajudam a encontrar defeitos antes que se passe à fase seguinte
  - utilizam uma técnica de leitura, buscando a localização de erros ou defeitos, de acordo com um critério pré-estabelecido

© KKB/PUC-Rio

Documento de Requisitos: inspeções

- Algumas técnicas para leitura:
  - Ad hoc (baseada na experiência individual)
  - Checklist (lista de ítens a serem verificados)
  - Perspective-based reading (baseada em diferentes visões ou perspectivas)

© KKB/PUC-Rio

Documento de Requisitos: inspeção

- ETAPAS DA INSPEÇÃO
  - Planejamento: seleção de participantes, atribuição de papéis, agendamento da reunião de inspeção, reprodução e distribuição do material a ser utilizado
  - Visão geral: autor apresenta o artefato a ser inspecionado aos participantes
  - Inspeção: inspetores avaliam o artefato e registram defeitos encontrados
  - Coleção: defeitos são reunidos e comunicados ao autor do artefato
  - Correção: defeitos encontrados são corrigidos
  - Acompanhamento: verificação da correção dos defeitos apontados

© KKB/PUC-Rio

Documento de Requisitos: inspeção

393

PAPÉIS:

- organizador: responsável pela organização e condução do processo como um todo
- autor (do artefato): apresenta uma visão global do mesmo antes da inspeção ter início
- inspetor: analisa os artefatos, seguindo uma técnica de leitura pré-definida, anotando os defeitos encontrados
- secretário: registra a reunião, reúne os defeitos encontrados pelos vários inspetores e os consolida num único documento
- relator: apresenta os defeitos coletados aos autores do artefato
- moderador: responsável por mediar e resolver eventuais conflitos

© KKB/PUC-Rio

Documento de Requisitos: inspeções

394

- Inspeções *ad hoc*:
  - baseadas fortemente no conhecimento e na experiência do inspetor
  - critérios e métodos adotados na análise/leitura dos artefatos dependem do inspetor que as efetua

© KKB/PUC-Rio

Documento de Requisitos: inspeções

395

- Algumas qualidades identificáveis por esta técnica:
  - clareza (os requisitos estão bem especificados?)
  - completude (estão presentes todos os requisitos necessários à especificação do sistema?)
  - consistência (os requisitos são consistentes com a visão geral do sistema

© KKB/PUC-Rio

Documento de Requisitos: inspeções

396

- Inspeções *ad hoc*: qualidades identificáveis ...
  - corretude (os requisitos descrevem as funcionalidades de maneira correta?)
  - funcionalidade (as funcionalidades descritas são necessárias e suficientes para atingir os objetivos do sistema?)

© KKB/PUC-Rio

Documento de Requisitos: inspeções

397

- Inspeções *ad hoc*: qualidades identificáveis ...
  - testabilidade (as funcionalidades permitem a verificação ou teste de forma a mostrar que os requisitos são satisfeitos?)
  - detalhamento (o nível de detalhe nos requisitos é suficiente para fornecer uma base adequada ao *design* do sistema?).

© KKB/PUC-Rio

Documento de Requisitos: inspeções

398

- Inspeções baseadas em *check lists*:
  - inspetores utilizam uma lista com os itens a serem verificados
  - cada artefato tem uma lista específica (Documento de Requisitos, Casos de Uso, Cenários, Glossário, Léxico, ...)

© KKB/PUC-Rio

Documento de Requisitos: inspeções

399

- Inspeções baseadas em *check lists*:
  - os defeitos são anotados no artefato sendo analisado
  - após a revisão, é realizada uma reunião onde os problemas encontrados são relatados aos desenvolvedores

© KKB/PUC-Rio

Documento de Requisitos: inspeções

400

- Defeitos que podem ser detectados:
  - sintaxe incorreta nos artefatos
  - informação inconsistente entre artefatos (requisitos e casos de uso, requisitos e glossários)
  - requisitos não funcionais não explicitados

© KKB/PUC-Rio

Documento de Requisitos: inspeções

401

- Defeitos que podem ser detectados:
  - informação ambígua (símbolos, termos ou sentenças que possam provocar diferentes interpretações)
  - informação desnecessária (por exemplo, atores em excesso)
  - ausência de informação (por exemplo, atores, pré ou pós-condições em casos de uso)
  - exceções não previstas

© KKB/PUC-Rio

Variação

402

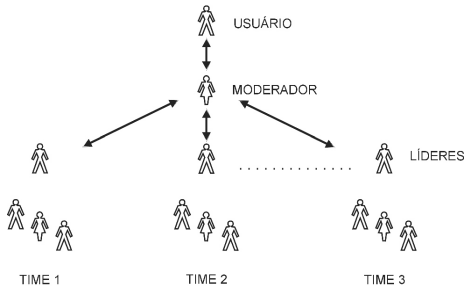
- N-Fold Inspection
  - vários times
  - cada um conduz um processo de inspeção independente
  - comparação dos resultados
  - relatório final

© KKB/PUC-Rio

Figura N-fold

403

N-FOLD INSPECTION



UM DOCUMENTO É REVISTO POR N TIMES, ONDE CADA TIME USA O PROCESSO DE INSPEÇÃO PARA DESCOBRIR ERROS.

© KKB/PUC-Rio

Paralelo é melhor

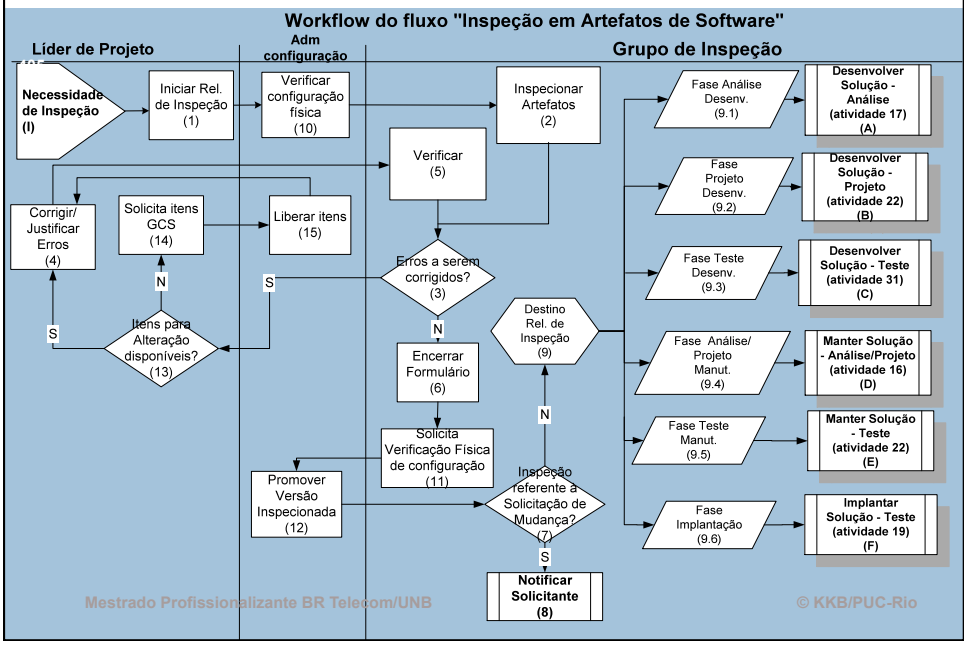
404

- Múltiplos times de inspeção acham mais defeitos do que um único time maior
- Os times tendem a achar sub conjuntos de defeitos diferentes, indicando:

A combinação dos resultados dos vários times tem um caráter aditivo e não redundante.

© KKB/PUC-Rio

Processo Petrobrás (realizar inspeção - Sinpep)


Workflow do fluxo "Inspeção em Artefatos de Software"
Líder de Projeto
Adm configuração
Grupo de Inspeção
Necessidade de Inspeção (I)
Iniciar Rel. de Inspeção (1)
Verificar configuração física (10)
Inspecionar Artefatos (2)
Verificar (5)
Corrigir/ Justificar Erros (4)
Solicita itens GCS (14)
Liberar itens (15)
Erros a serem corrigidos? (3)
Itens para Alteração disponíveis? (13)
Encerrar Formulário (6)
Destino Rel. de Inspeção (9)
Solicita Verificação Física de configuração (11)
Promover Versão Inspecionada (12)
Inspeção referente a Solicitação de Mudança? (7)
Notificar Solicitante (8)
Fase Análise Desenv. (9.1)
Fase Projeto Desenv. (9.2)
Fase Teste Desenv. (9.3)
Fase Análise/ Projeto Manut. (9.4)
Fase Teste Manut. (9.5)
Fase Implantação (9.6)
Desenvolver Solução - Análise (atividade 17) (A)
Desenvolver Solução - Projeto (atividade 22) (B)
Desenvolver Solução - Teste (atividade 31) (C)
Manter Solução - Análise/Projeto (atividade 16) (D)
Manter Solução - Teste (atividade 22) (E)
Implantar Solução - Teste (atividade 19) (F)
N
S
Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio

Processo Petrobras

406

1. Iniciar Relatório de Inspeção:

a. preenchido pelo líder do projeto:informar a fase, a lista de artefatos, o critério de seleção dos itens de inspeção e a data esperada de conclusão.

2. Inspecionar Artefatos

2.a. Organizar - organizador Relatório de Inspeção em Artefatos de Software → indica o Inspetor; poderá ser alocado mais de um Inspetor para participar do processo.

2.b. Conferir – Inspetor verifica se os itens de configuração estão disponíveis. Nos casos de manutenção em projetos de software → artefatos acrescentados e/ou modificados.

2.c. Inspecionar –

inspeção baseada nos critérios de seleção definidos no relatório

guiando-se pelo Mapa de Inspeção e Lista de Verificação.

registrar itens de melhoria no Relatório

classificação dos erros:

- Erro crítico: compromete a qualidade do produto do projeto.
- Erro maior: compromete a qualidade do processo.
- Erro menor: compromete padrões de nomenclatura da organização.
- Comentários: não comprometem a qualidade, mas indicam melhorias que podem ser implementadas.

3. 3. Erros a serem corrigidos?:

a. Inspetor envia o Relatório de Inspeção em Artefatos de Software ao Líder do Projeto para que seja providenciada a correção.

Mestrado Profissionalizante BR Telecom/UNB © KKB/PUC-Rio

Processo Petrobras

407

5. **Verificar:**

a. Inspetor verifica se os erros encontrados na inspeção foram corrigidos

b. Caso novos itens de melhoria sejam encontrados durante a verificação, estes são informados no Relatório

6. **Encerrar Formulário:** Concluir o Relatório de Inspeção em Artefatos de Software

7. **Inspeção referente à Solicitação de Mudança?:**

a. Se a inspeção for iniciada a partir da saída do fluxo Tratar Mudança, o Inspetor informa ao solicitante que a inspeção foi concluída;

b. caso contrário, o relatório é enviado ao Líder, na fase do projeto que originou a solicitação de inspeção.

8. **Notificar Solicitante:** Caso a inspeção tenha sido iniciada a partir da saída do fluxo Tratar Mudança o Líder de Projeto é informado que a inspeção foi concluída.

9. **Definir Destino do Relatório de Inspeção:**

Inspetor define o destino do Relatório de Inspeção em Artefatos de Software em função de:

Ponto de Inspeção (Análise, Projeto e Validação) e

Processo (Desenvolver Solução, Manter Solução ou Implantar Solução) em que foi solicitada a inspeção.

© KKB/PUC-Rio

Pontos de inspeção

408

Pontos de Inspeção	Artefatos a Inspecionar	Obrigatório (S/N)	Localização do modelo (template)	Localização do padrão associado	Ferramentas
Análise	Diagrama de caso de uso	S	N/A	Referência [1]	Rational Rose
	Documento de visão	S	visao.dot (NF-11-00060-0)	-	MS-Word
	Especificações suplementares	S	espec_sup.dot (NF-11-00058-0)	-	MS-Word
	Especificação do caso de uso	S(*7)	caso_de_uso.dot (NF-11-00057-0)	Referência [1]	MS-Word
	Glossário	S	glossario.dot (NF-11-00059-0)	-	MS-Word

© KKB/PUC-Rio

Desafios da inspeção

409

- Grandes documentos de requisitos
  - revisões informais e incrementais durante o desenvolvimento da especificação
  - cada inspetor começa em um ponto diferente da especificação
  - dividir em vários times pequenos - cada um inspeciona uma porção do documento

© KKB/PUC-Rio

Desafios da inspeção

410

- Times de inspeção muito grandes
  - dificuldade de se marcar reuniões
  - conversas paralelas
  - difícil chegar a um consenso
- o que fazer?
  - ter certeza de que os participantes estão lá para inspecionar e não para "espionar" a especificação ou manter uma posição política

© KKB/PUC-Rio

Desafios da inspeção

411

- Times de inspeção muito grandes
  - Entender que ponto de vista (freguês, usuário, desenvolvedor…) do inspetor e manter apenas um para cada parte interessada
  - Estabelecer vários times menores e fazer a inspeção em paralelo. Combinar as listas e remover duplicatas

© KKB/PUC-Rio

Desafios da inspeção

412

- Distância geográfica dos inspetores
  - videoconferência, teleconferência, e-mail, web
    - difícil de observar linguagem corporal e expressões, especialmente em teleconferência por mail
    - difícil de moderar
  - redução de 25% de efetividade
    - [Wiegers98] - The seven deadly sins of software reviews - Software Development -6(3) pp.44-47

© KKB/PUC-Rio

Requistos & Certificação

CMM – Nível 2

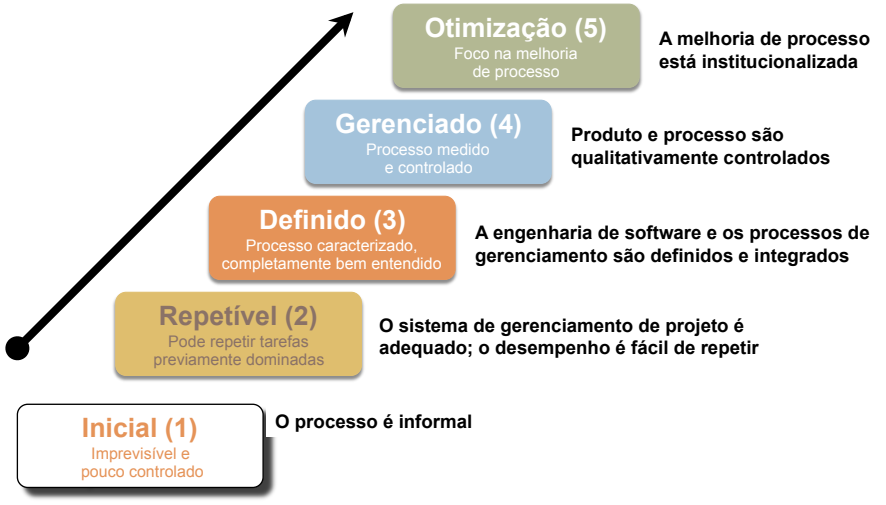
Áreas chaves do processo

- Gerência de Requisitos
- Planejamento de Projetos
- Acompanhamento de Projetos
- Gerenciamento de Subcontratação
- Garantia da Qualidade de Software
- Gerência de Configuração

Fonte – SEI – Mark Paulk

©KKB/PUC Rio

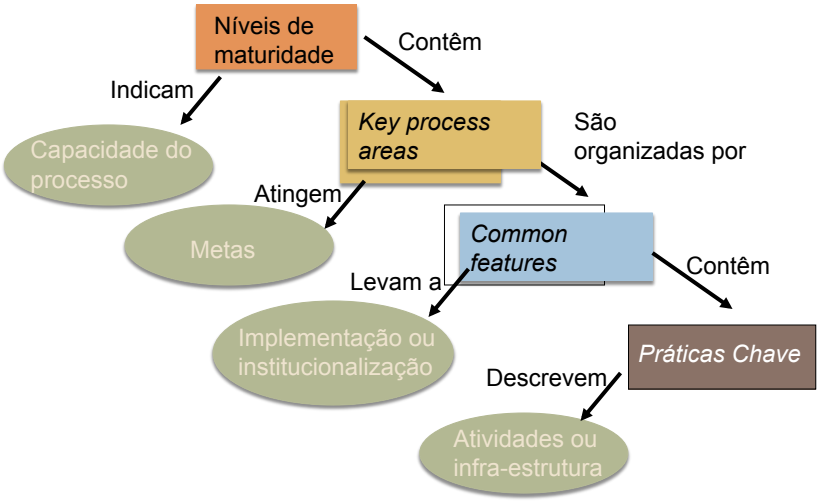
Requisitos & CMM


Otimização (5)
Foco na melhoria de processo
A melhoria de processo está institucionalizada
Gerenciado (4)
Processo medido e controlado
Produto e processo são qualitativamente controlados
Definido (3)
Processo caracterizado, completamente bem entendido
A engenharia de software e os processos de gerenciamento são definidos e integrados
Repetível (2)
Pode repetir tarefas previamente dominadas
O sistema de gerenciamento de projeto é adequado; o desempenho é fácil de repetir
Inicial (1)
Imprevisível e pouco controlado
O processo é informal

Fonte – SEI – Mark Paulk

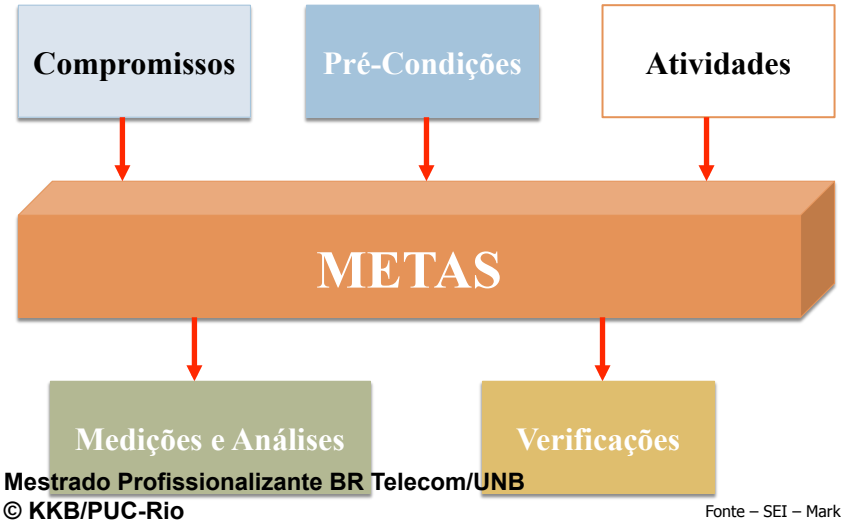
©KKB/PUC Rio

Estrutura do CMM



Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio
Fonte – SEI – Mark Paulk

Estrutura das Áreas-Chave de Processo



Mestrado Profissionalizante BR Telecom/UNB
© KKB/PUC-Rio
Fonte – SEI – Mark Paulk

Práticas Chave

417

- Descrevem "o que" será feito para cada Área-Chave do Processo, mas não "como"
  - São requisitos a serem atendidos

Meta Gerência de Requisitos

418

- Controle

Os requisitos de software devem ser controlados para uso gerencial.

©JKB/PUC Rio

Fonte – SEI – Mark Paulk

Meta Gerência de Requisitos

419

- Consistência

Os planos, produtos e atividades de software devem ser mantidos consistentes com os requisitos de software.

©KKB/PUC Rio

Fonte – SEI – Mark Paulk

Meta Gerência de Requisitos

420

- Compromisso

Seguir política organizacional para gerenciar os requisitos de software.

- Documentar requisitos
- Participação/Revisão dos requisitos pelos interessados
- Manter consistência entre artefatos e requistos

©KKB/PUC Rio

Fonte – SEI – Mark Paulk

Meta Gerência de Requisitos

421

- Pré Condições
  - 1. **analisar os requisitos** de sistema e **alocá-los** ao hardware, software ou outros componentes
  - 2. documentar requistos
  - 3. disponibilizar recursos necessários para gerenciar os requisitos de software
  - 4. treinar equipe para realizar as atividades de gerência de requisitos

Fonte – SEI – Mark Paulk

© KKB/PUC-Rio

Meta Gerência de Requisitos

422

Fonte – Claudia Hazan

- Atividades I

revisar os requisitos de software, antes de incorporá-los ao projeto

- **Identificar requisitos incompletos ou ausentes**
- **Determinar se os requisitos estão claros,**
- **possíveis de serem implementados, consistentes e verificáveis**
- **Revisar requisitos com problemas potenciais**
- **Negociar compromissos com os grupos envolvidos**

© KKB/PUC-Rio

Meta Gerência de Requisitos

423

Fonte – Claudia Hazan

- Atividades II

Utilizar os requisitos alocados como base para as atividades do desenvolvimento de software.

- **Gerenciados e controlados**
- **Base para o desenvolvimento dos requisitos de Sw**
- **Base para o plano de desenvolvimento de Sw**
- **Base para o Projeto do Sw.**

© KKB/PUC-Rio

Meta Gerência de Requisitos

424

- Atividades III

Revisar alterações nos requisitos alocados e incorporá-las ao projeto de software

- **Revisar com a gerência sênior alterações nos compromissos feitos com grupos externos.**
- **As alterações de compromissos feitos dentro da organização são negociadas com os grupos envolvidos.**

O impacto nos compromissos existentes é avaliado e mudanças são negociadas, quando apropriado.

© KKB/PUC-Rio

Fonte – SEI – Mark Paulk

Alterações nos Requisitos

425

As alterações que precisam ser feitas nos planos de software, artefatos e atividades resultantes da alteração dos requisitos são:

- **- Identificadas**
- **- Avaliadas**
- **- Avaliadas sob o ponto de vista de risco**
- **- Documentadas**
- **- Planejadas**
- **Comunicadas aos grupos e indivíduos**
- **envolvidos**
- **- Acompanhadas até a finalização**

© KKB/PUC-Rio

Fonte – SEI – Mark Paulk

Exemplo

426

	Tecnologia da Informação Gerência Provedora

Registro de Solicitação de Mudança		
Solicitante: Nome do usuário que solicitou a mudança.	**Unidade: Sigla da unidade organizacional.**	**Contato: Fone/Correio**
Tipo de Mudança		
Alteração	**Cancelamento**	
Funcional	**Não Funcional**	
( ) Novo Requisito ( ) Exclusão de Requisito ( ) Alteração de Requisito *[Permitir mais de uma seleção.]*	( ) Referente a Infra-estrutura ( ) Referente ao Apoio ( ) Outros. Especifique _____	

© KKB/PUC-Rio

Especificação Projeto
Origem: *[marcar a opção que retrate a origem da solicitação de mudança.]* ( ) Necessidade do Cliente ( ) Necessidade interna
Descrição: *[Descrição detalhada da mudança]*
Justificativa: *[Descrição dos motivos e justificativas para a solicitação da mudança.]*
Requisitos: *[Descrição dos requisitos necessários para a execução da solicitação de mudança.]*
Análise de Impacto Projeto: ***[Descrever a análise de impacto das alterações propostas, principalmente as de alto impacto e que envolvem ações inter-áreas.Preenchido pelo Líder de Projeto]]***
Ação Proposta Projeto
[Encaminhamento para atendimento da mudança.]
Especificação Produto
Análise de Impacto Produto: ***[Descrever a análise de impacto das alterações propostas, relacionadas com as fases e tarefas. Preenchido pelo Líder de Produto]***
Fase da Ocorrência: *[marcar a opção que retrate a fase onde incidiu a abertura desta solicitação.]*

( ) Desenvolvimento da solução	( ) Requisitos ( ) Análise ( ) Projeto ( ) Implementação ( ) Teste	*[Opção só ativada se a opção do usuário for desenvolvimento da solução ou manutenção da solução.]*
( ) Manutenção da solução		
( ) Implantação da solução		

Ação Proposta Produto
[Encaminhamento para atendimento da mudança.]

Meta Gerência de Requisitos

- Medição e análise

Objetivo:
Definir e utilizar medições que determinem o estado das atividades de gerência de requisitos

Meta Gerência de Requisitos

429

- Exemplos:
  - **O estado de cada um dos requisitos alocados**
  - **As atividades de alteração para cada um dos requisitos alocados.**
  - **O número acumulado de alterações para cada um dos requisitos alocados, incluindo o número total de alterações propostas, abertas, aprovadas e incorporadas a baseline do sistema.**

© KKB/PUC-Rio

Meta Gerência de Requisitos

430

Fonte – Claudia Hazan

- Verificação
  - A gerência sênior revisa, periodicamente, todas as atividades de gerência dos requisitos de software
  - O gerente do projeto revisa, periodicamente e quando da ocorrência de eventos, todas as atividades de gerência dos requisitos do software.

© KKB/PUC-Rio

Meta Gerência de Requisitos

431

VERIFICAÇÃO

- O grupo de GQS revisa e/ou audita as atividades e artefatos utilizados para gerenciar os requisitos alocados, reportando seus resultados.

© KKB/PUC-Rio

Meta Gerência de Requisitos

432

VERIFICAÇÃO

- São realizadas, no mínimo as seguintes verificações:
  - Revisão dos requisitos de software
  - Planos de software
  - Artefatos e produtos de trabalho
  - Atividades da gerência de requisitos
  - Comprometimento dos grupos envolvidos

© KKB/PUC-Rio